De-interleaving of Radar Pulses for EW Receivers with an ELINT Application

By

Mohammad Aldossary

Department of Electrical Engineering

Supervisor

Professor Michael Inggs

Department of Electrical Engineering

Cape Town, March 2017

A dissertation submitted to the Department of Electrical Engineering at the University of Cape Town, in fulfilment of the requirements for the degree of
Master of Science in Electrical Engineering

De-interleaving of Radar Pulses for EW Receivers with an ELINT Application

Mohammad Aldossary

October 20, 2017

Declaration

I know the meaning of plagiarism, and I declare that, this dissertation is completely my personal work without any support from any external party. The dissertation is being submitted for the degree of Master of Science in Engineering at the University of Cape Town. The dissertation has not been submitted before for any degree or examination in any other university.

.

.

.

.

Signature of Author .....

Abstract

De-interleaving is a critical function in Electronic Warfare (EW) that has not received much attention in the literature regarding on-line Electronic Intelligence (ELINT) application. In ELINT, on-line analysis is important in order to allow for efficient data collection and for support of operational decisions. This dissertation proposed a de-interleaving solution for use with ELINT/Electronic-Support-Measures (ESM) receivers for purposes of ELINT with on-line application. The proposed solution does not require complex integration with existing EW systems or modifications to their sub-systems. Before proposing the solution, on-line de-interleaving algorithms were surveyed. Density-based spatial clustering of applications with noise (DBSCAN) is a clustering algorithm that has not been used before in de-interleaving; in this dissertation, it has proved to be effective. DBSCAN was thus selected as a component of the proposed de-interleaving solution due to its advantages over other surveyed algorithms. The proposed solution relies primarily on the parameters of Angle of Arrival (AOA), Radio Frequency (RF), and Time of Arrival (TOA). The time parameter was utilized in resolving RF agility. The solution is a system that is composed of different building blocks. The solution handles complex radar environments that include agility in RF, Pulse Width (PW), and Pulse Repetition Interval (PRI).

Contents

List of Figures

List of Tables

Nomenclature

AAM	Air-Air-Missiles
ADC	Analog-to-Digital-Converter
ANN	Artificial Neural Networks
AOA	Angle of Arrival
APNN	Adaptive Probabilistic Neural Network
ARM	Anti Radiation Missile
ARW	Anti-Radiation Weapons
ASIC	Application-Specific Integrated Circuit
ASM	Anti-Ship Missile
ASP	Antenna Scan Period
BSS	Blind Source Separation
CAM	Content Addressable Memory
CCDM	Concurrent Classification and De-interleaving Machine
CCL	Cone Cluster Labeling
CDBW	Composed-Density-Between-and-Within-Clusters
CDIF	Cumulative Difference Histogram
CMSVC	Cone Mapping Support Vector Clustering
COMMINT	Communication Intellegence
CPU	Central Processing Unit

CSSI	Cumulative Square Sine-Wave Interpolation
CSSI	Cumulative Square Sine-wave Interpolating
CW	Continuous Wave
DBSCAN	Density-Based Spatial Custering of Applications with Noise
DCSVC	Delaminating coupling SVC
DEW	Directed Energy Weapons
DF	Direction Finding
DIFM	Digital Instantaneous Frequency Measurement
DPU	Data Processing Unit
DSOM	Dynamic Self-Organizing Map
DSP	Digital Signal Processing
DTOA	Differential Time of Arrival Histogram
EA	Electronic Attack
ECCM	Electronic Counter Counter Measures
ECM	Electronic Counter Measures
ELINT	Electronic Intellegence
EM	Electro-Magnetic
EMS	Electro-magnetic Spectrom
EOB	Electronic Order of Battle
EP	Electronic Protection
ES	Electronic Support
ESM	Electronic Support Measures
EW	Electronic Warfare
F-ART	Fuzzy-ART

FFT	Fast Fourier Transform
FMMC	Fuzzy-Min-Max clustering
FPGA	Field Programmable Gate Array
FT	Fourier Transform
HMI	Human Machine Interface
HOS	Higher Order Statistics
HPBW	Half-Power-Beam-Width
I/Q	In-phase and Quadrature
IAFC	Integrated Adaptive Fuzzy Clustering
IBW	Instantaneous Band-Width
ICA	Independent Component Analysis
IF	Intermediate Frequency
IFM	Instantaneous Frequency Measurement
ISO-DATA	Iterative Self Organizing Data Analysis
KFCM	Kernel Fuzzy C-Means
Kpps	Kilo pulse per second
LFM	Linear Frequency Modulation
LO	Local Oscillator
LPI	Low Probability of Intercept
MDL	Minimum Description Length
MFR	Multi-Function Radars
MOP	Modulation-ON-Pulse
MP	Matching Pursuit
MTI	Moving Target Indicator

NLFM	Nonlinear Frequency Modulation
OISVM	On-line Independent Support Vector Machines
OpELINT	Operational ELINT
PA	Pulse Amplitude
PC	Personal Computer
PDW	Pulse Descriptor Word
POI	Probability-of-Intercept
PPV	Pulse Parameter Vector
PRF	Pulse Repetition Frequency
PRI	Pulse Repetition Interval
PW	Pulse Width
QPSK	Quadri-phase shift keying
RAM	Random Access Memory
RF	Radio Frequency
RMS	Root Mean Square
RNN	Recurrent Neural Networks
RST	Rough Set Theory
SAM	Surface-Air-Missiles
SCA	Sparse Component Analysis
SDIF	Sequential Difference Histogram
SEI	Specific Emitter Identification
SIGINT	Signal Intelligence
SNR	Signal-to-Noise-Ratio
SOFM	Self Organizing Feature Map

SS	Sequence Search
SVC	Support Vector Clustering
TDH	Time Difference Histogram
TDOA	Time-Difference-of-Arrival
TechELINT	Technical ELINT
TOA	Time of Arrival
VLSI	Very-Large-scale Integration

Chapter 1

Introduction

1.1 Background

Radar in modern warfare is a tool that utilizes the electromagnetic spectrum (EMS) to implement its functions. Important radar functions include searching, tracking, guidance, imaging, and navigation. Electronic Warfare (EW) is the science of reacting to EMS threats, which are mainly radar and communication threats. EW consists of three related sectors, viz., Electronic Protection (EP), Electronic Attack (EA), and Electronic Support (ES). EA uses the EMS offensively to disable the enemy's capabilities. EP uses the EMS defensively in order to protect friendly capabilities. ES – the focus of this dissertation – is responsible for supplying intelligence and threat recognition that are necessary for EA and EP to function effectively.

ES utilizes special receivers to intercept electromagnetic (EM) emissions of radars. Such EM emissions of radars are mostly formed (modulated) in the shape of pulsed signals. In order to gather useful information, the characteristic parameters of intercepted signals are measured. Some parameters of radar signals can be obtained from a single pulse, whereas other parameters can only be obtained from a greater number of pulses. Radio Frequency (RF), Pulse Width (PW), Time of Arrival (TOA), Angle of Arrival (AOA), and Pulse-Amplitude (PA) are common examples of parameters obtained from a single pulse. This set of parameters is referred to herein as a Pulse Parameter Vector (PPV). Each measured pulse, therefore, is represented by a single PPV. Practically, parameters that require a greater number of pulses in order to be measured make use of multiple PPVs in the measurement process; they are referred to as derived parameters. An obvious example of such a derived parameter is the pulse-repetition-interval (PRI), which is the parameter that is obtained from subtracting the TOA parameter of two consecutive pulses (or PPVs). Another

example is the antenna scan period (ASP), which is measured by utilizing multiple PPVs.

The environment, where the interception takes place, usually consists of many emitters. The measured PPVs in this situation will thus belong to pulses from multiple emitters. This create two problems: Firstly, although there are many measured parameters (PPVs) that belong to different emitters, it is not known to which emitter each of these PPV belongs. Secondly, a sequence of pulses (PPVs) from the same emitter is required in order to extract further important parameters (such as PRI and ASP), which are essential for important EW tasks (such as the classification and identification of emitters). In order to solve these two problems, it becomes important to separate (sort) interleaved emitters' pulses (PPVs) into distinctive groups. This sorting process is called de-interleaving.

De-interleaving allows for further analysis of emitter parameters, particularly for the measurement of inter-pulse parameters, which (in addition to PPV parameters) are important for classification of an emitter. In classification, the type of emitter (or the group to which a specific emitter belongs) is determined. In other words, classification answers the question of: how is the intercepted emitter related to other emitters in the threat library? Multiple radars that belong to the same manufacturer and that may even have the same model may exist in the environment. These radars are usually classified into the same emitter class. In order to distinguish radars of the same class from each other, an identification task should take place. More parameters are required in order to perform the identification process. These parameters are usually related to imperfections in the manufacturing process of radar components, such as local oscillator, power amplifiers, etc. Pulse rise-time, and phase coherency with time are examples of parameters that can be used in identification. Measurement processes for parameters that are used for identification are usually more complex compared to PPV parameters. Furthermore, parameters that are suitable for identification of one type (or class) of radars are not necessary suitable for other classes of radars. Hence, it is impractical to measure these parameters before completing the classification process.

De-interleaving is used in EW within ES. In the context of radar, ES consists of two divisions: Electronic Intelligence (ELINT), and Electronic Support Measures (ESM). ESM receivers are linked with radar warning system or radar countermeasure systems in order to react to encountered threats. Hence, time is a critical factor in ESM. The goal of ELINT, in contrast, is to create threat databases that are used by ESM (among other users), and to update the Electronic Order of Battle (EOB) which is concerned with movements, location updates, and the readiness of adversary emitters.

With advances in digital hardware technology, it becomes possible (and it is sometimes preferred) to use the same hardware (i.e. receiver) for both ESM and ELINT. However, the requirements of

de-interleaving are different for ESM and ELINT, although they use same hardware. In ELINT, on-line analysis is important in order to allow for efficient data collection and to support operational decisions. However, post analysis can be performed off-line on recorded raw data (i.e. recorded PPVs). As have been discussed above, the objectives and constraints of ELINT and ESM are different, and consequently the requirement for de-interleaving, classification, and identification are different too.

1.2 Objectives

The aim of this dissertation is to propose an appropriate de-interleaving algorithm for use with ELINT/ESM receivers for purposes of ELINT (with on-line application).

An appropriate de-interleaving algorithm is defined in this dissertation as; an algorithm that satisfies the following requirements:

- It must have the capacity to handle a complex signal environment, which is an environment that includes emitters with frequency agility, pulse-width agility, jittered PRI, and staggered PRI.
- It must have the capacity to sort a high number of active emitters, which is defined herein as 20 emitters or more. Active emitters are those intercepted by ESM/ELINT receiver during an ELINT mission; they exist in the interleaved pulse train buffer.
- It must have the ability to handle dense pulse environment. Dense pulse environment is defined as environment with average pulse rate of 10 k pulse per second or more.
- It must requires minimal hardware upgrades, and it may not impose major hardware upgrades on current systems. Major hardware upgrades are defined as upgrades that involve subsystems that require complex integration with the existing ESM/ELINT receiver, or with the platform, for example, changing the Direction Finding (DF) subsystem, or adding more digital, or RF channels.
- It may not impose higher resource consumptions on existing ESM/ELINT signal processing unit than that used on the system of interest. This unit usually includes a large Field Programmable Gate Array (FPGA), wheres large portion of its resources is already utilized.
- It may not be sensitive to missing pulses, which are defined as the pulses missing from intercepted sequence.

Based on the above introduction, the objectives of this dissertation are:

- Firstly, to survey the literature for de-interleaving algorithms that can be used for on-line ELINT application.
- Secondly, to propose an appropriate de-interleaving algorithm for on-line ELINT application.
- Thirdly, to evaluate proposed algorithm by mean of computer simulation.

1.3 Approach

This dissertation will adopt the following approach:

- It will start by reviewing the literature regarding the introduced de-interleaving requirements introduced above.
- Thereafter, data of interleaved pulse train (intercepted by ELINT/ESM receiver) will be simulated using Matlab.
- And finally, the proposed de-interleaving algorithm will be modelled and simulated using Matlab.

1.4 Assumptions

The following assumptions apply:

- Emitters and the EW receiver are stationary.
- A non-cumulative time jitter model is assumed.
- All radars are scanning type radars.
- Emitters intercepted from the same direction operate on different frequencies (radars within 10^o separation in angle will be considered to be on the same direction). This assumption is made because scanning radars may have line-of-sight in this case and may use different frequencies in order to avoid interference.
- The EW receiver measures the following radar parameters: TOA, RF, PW, PA, and AOA, which are the minimum parameters that should be measured by modern EW receivers.

- De-interleaving is applicable for pulsed radars only. This assumption is made because the de-interleaving is performed based on Pulse Descriptor Word (PDW). Some receivers may not be able to provide PDW for non-pulsed radars, such as Low Probability of Intercepts (LPI) radars or Continuous Wave (CW) radars.
- The rate of maximum missing pulses for each emitter is 1%. This assumption is made in order to bound the problem of the de-interleaving.
- Latency in ELINT is not a critical factor. This assumption agrees with the nature of ELINT missions, which do not require instant reaction to threat's signals.
- All emitters are unknown (there will be no prior knowledge about emitters). This assumption is made to represent the worst case scenario for an ELINT mission regarding the known emitters.

The first four assumptions are made in order to create boundaries for the addressed de-interleaving problem in order to fit the time frame of the dissertation.

1.5 Scope

Due to restrictions in obtaining real data, only simulated data are used in this dissertation. The scope of the dissertation is depicted in Figure (1.1).

1.6 Motivation

For ELINT applications, when de-interleaving is addressed in the literature, it is usually addressed as part of off-line (post-analysis) applications. Conversely, for on-line de-interleaving applications, the literature is mostly targeting ESM. De-interleaving is a critical function in EW that has not received much attention in the literature regarding on-line ELINT applications.

On-line de-interleaving for ELINT allows for more efficient data collection during a data gathering mission. The outcome of such a data-gathering mission definitely affects the post-processing analysis.

In fact, advanced EW hardware becomes useless or unreliable without proper de-interleaving that suits its function. The objectives of ELINT and ESM are different, and consequently their respective on-line de-interleaving requirement are different too. Latency, for instance, is a critical factor in ESM, and hence, real-time performance is an ultimate goal for most algorithms in the literature that is targeting on-line de-interleaving. For ELINT, in contrast, latency is not a critical

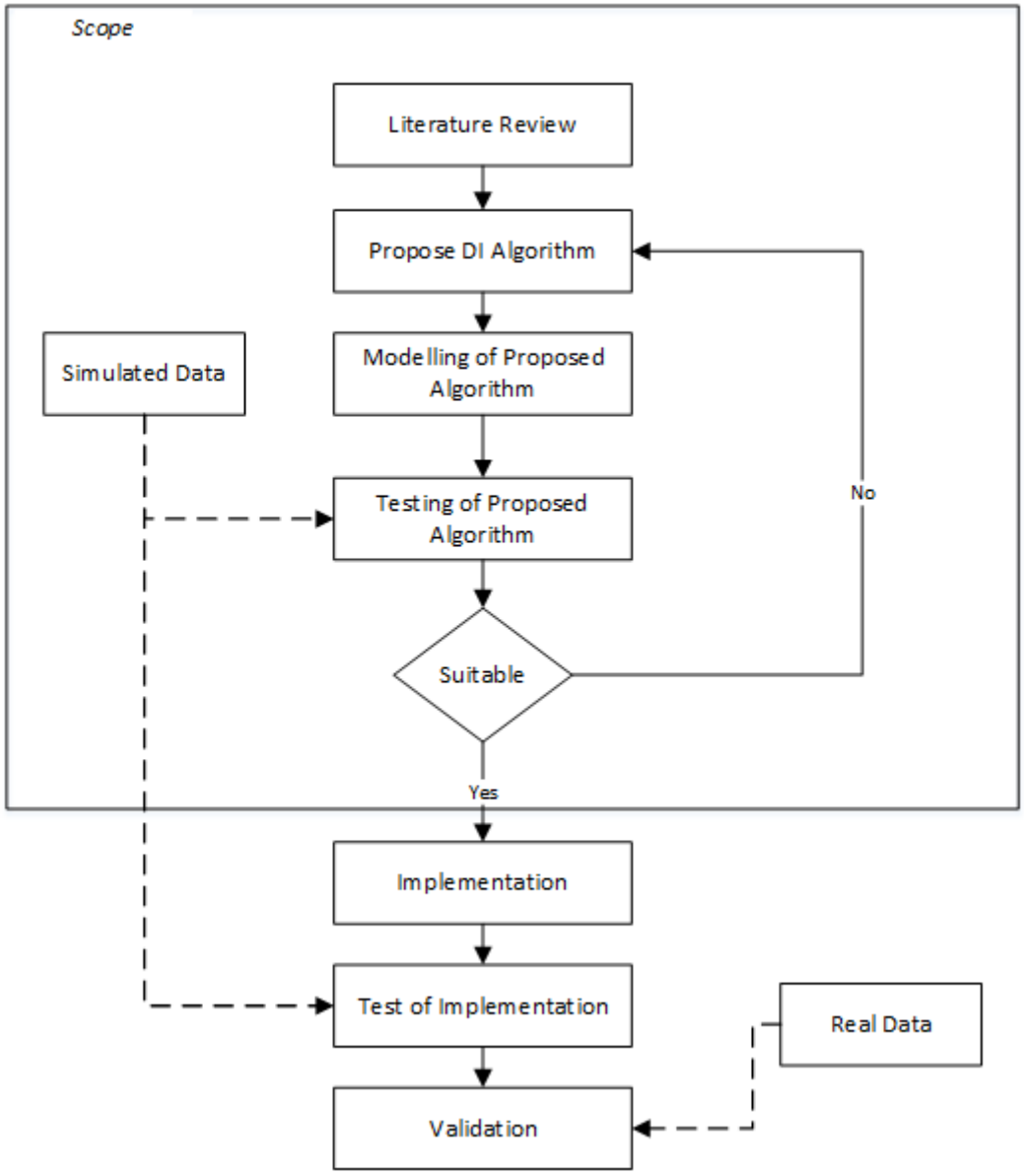


Figure 1.1: Scope of dissertation.

factor. However, we need to deal with dense emitter environment where high pulse rates are expected.

Realistic ELINT scenarios regarding hardware constraints or environments need to be considered in the used de-interleaving algorithms. Using same EW receiver for ELINT and ESM becomes more common in practice due to advances in technology, in addition to other factors related to the cost of missions. It is desired in this dissertation to propose a de-interleaving algorithm in order to work with available ESM/ELINT receivers without imposing requirements for major hardware upgrades.

Given the reasons above, the literature will be surveyed to identify the available algorithms. These algorithms will be reviewed regarding the desired ELINT application in order to select or propose an appropriate algorithm for our purposes. This review will also be useful for researchers who are interested in studying de-interleaving algorithms for on-line ELINT applications.

1.7 Related Background

In this section, some of important background terms such as Electronic Warfare, Radar Threats and Electronic Intelligence are introduced.

Electronic Warfare Electronic Warfare is defined as "a military action whose objective is to control the EM spectrum" [1]. EW consists of three divisions: Electronic Attack (EA), Electronic Support (ES), and Electronic Protection (EP). ES involves actions to "search for, intercept, identify and locate sources of intentional and unintentional radiated EM energy" [1]. EA involves using "directed energy to attack equipment, facilities, or personnel with the intent of degrading, neutralizing, or destroying enemy capability" [1]. EP involves actions "to protect, personnel, facilities, or equipment from the effect of friendly or enemy employment of EW that degrades, neutralize, or destroy friendly combat capability" [1]. These divisions of EW are illustrated in Figure (1.2).

In terms of its old classification, EW was divided into three sectors: Electronic Support Measures (ESM), Electronic Counter Measures (ECM), and Electronic Counter Counter Measures (ECCM). Anti-Radiation Weapons (ARW) and Directed Energy Weapons (DEW), which were not part of EW in terms of the old classification, have joined EW in the new classification under the EA sector [2]. ESM is equivalent to ES, and ECCM is equivalent to EP. ECCM performs what is called a "soft kill" for target threats, while ARW and DEW perform what is called "hard kill". EA consists of both soft and hard kills [3].

De-interleaving has an EA application. Some references consider Signal Intelligence (SIGINT) as part of ES while others consider it as a separate branch. Regarding radar signals, de-interleaving

is used in ES in both ELINT and ESM. De-interleaving is also required in the passive radar seeker, (or Anti-Radiation Missile (ARM)), which is an EA application [4].

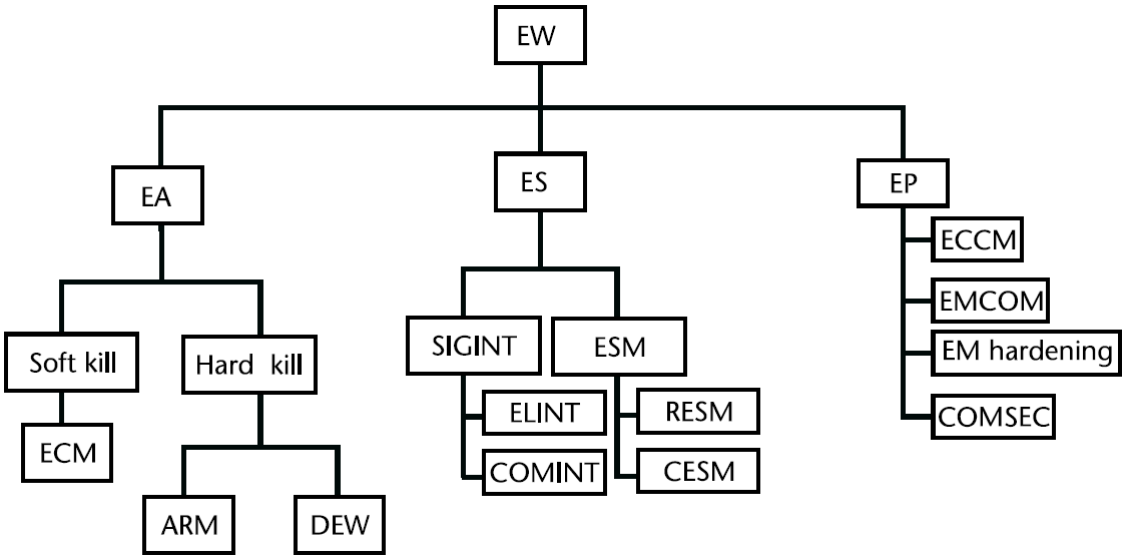


Figure 1.2: Divisions of EW [3].

Radar Threats There are many types of radar threats. The main ones include early warning radars, acquisition radars, target-tracking radars, and interceptor radars. Early warning radars are responsible for detecting targets at long ranges up to 400 km or more [3]. Although early warning radars have long range, they have limited resolution. Acquisition radars are similar to search radars, but with shorter range and better resolution; they search for targets for purpose of engagement, and then designate targets to specific tracking radars. Tracking radars provide fire control systems with accurate information about target location. Such targets can be engaged with missiles. Missiles can be active or semi-active: Active missiles have their own transmitter and receiver within the missile, while semi-active missiles contain only the receiver. A semi-active missile is dependent on an illuminating radar. Surface-air-missiles (SAM), Air-air-missiles (AAM), and Anti-ship missiles (ASM) are examples of radar guided missiles. Interceptor radars are used by fighter aircraft. These radars are usually multi-function radars (MFR) that perform search, track, guidance, imaging, and navigation functions. Low Probability of Intercept (LPI) radar is another radar threat that is not easy to detect using conventional methods. Each type of radar has characteristic signal parameters that suit its particular function. Early warning radars, for instance, use lower frequencies in order to allow for better propagation of EM waves, which allow them to reach long ranges. Intercepted radar signal parameters can provide a good idea about the function of the particular radar.

ELINT ELINT refers to the interception of non-communication EM signals in order to gain information about adversary systems [2]. ELINT consists of two divisions; Technical ELINT (TechELINT), and Operational ELINT (OpELINT) [5].

1.8 Organization of Dissertation

This section presents the structure of the dissertation and provides a summary of the main points covered in each chapter.

Chapter 2 De-interleaving algorithms process radar parameters that are measured by an EW receiver. Therefore, it is important to have some background information about these parameters and the system that measures them, before proceeding to discuss de-interleaving in particular. The background is important in order to assist us in making the right design decisions regarding the proposed de-interleaving solution.

This chapter gives some background about the most common pulse parameters that are measured by EW receivers. These parameters include those that are obtained from a single pulse, viz. TOA, AOA, RF, PW, and PA. These five parameters are usually constructing what is called a Pulse Descriptor Word (PDW) in EW receivers. In addition to these parameters, Modulation On Pulse (MOP) is another pulse parameter that becomes common in constructing the PDW of EW receivers. The common parameters also include PRI, which is not a part of PDW. PRI is an important parameter that is measured by EW systems and commonly used in de-interleaving. The background provided information about the mentioned parameters such as accuracy of measurement, in addition to some other information.

Various other parameters can be measured too such as Polarization, but it is rarely found in EW receivers due to unnecessary hardware complexity. There are other parameters that need de-interleaving in order to be obtained, such as ASP and Antenna Half-Power-Beam-Width (HPBW). These parameters can be used for the purposes of classification or identification but not for de-interleaving. The Pulse-rise-time parameter can be obtained (theoretically) from a single pulse. This parameter is used for purpose of identification and needs special measurement conditions. It is therefore unnecessary to discuss these latter parameters in this background, because they are not used for de-interleaving.

Chapter 2 also provides a brief background about the EW receiver of interest, in other words, the ELINT/ESM receiver. Important information about the receiver is introduced, such as frequency coverage, instantaneous bandwidth, configuration, and subsystems.

Chapter 3 This chapter surveys the literature for de-interleaving algorithms that can serve ELINT on-line application. In this review, the algorithms are divided into three categories: time-based algorithms, clustering algorithms, and combined algorithms.

Time-based algorithms rely primarily on the time of arrival (TOA) parameter in the de-interleaving of radar pulses. Theses algorithms use different techniques, such as the search technique, the histogram technique, and the domain transformation techniques. Some algorithms were found using one technique, while others were found using one primary technique, in addition to other secondary techniques.

The Sequence Search (SS) algorithm has a simple concept, and it was used with many subsequent algorithms. The SS algorithm, with some enhancement, is presented in [6], and [7]. Also, Recurrent Neural Networks (RNN), based on the state space model of the radar sequence was proposed for use in SS algorithm by [8].

Histogram techniques include the Time Difference Histogram (TDH) [9] [10], the Cumulative Difference Histogram (CDIF) [11], and the Sequential Difference Histogram (SDIF) [12]. Harmonics, jitter, and a dense radar environment are the main problems in histogram methods.

PRI-transform algorithms try to overcome the harmonic and jitter problems associated with histogram algorithms, as in [13], [14], [15], [16], and [15], but jitter still remains a challenge for these algorithms.

Time-Period PRI Transform (PRI-Map) by [17], is an extension of the PRI-transform algorithm in order to allow for detection of short time pulse trains. This algorithm is relatively complex compared to original PRI-transform algorithms.

Transform algorithms include the Hough Transform, which is an image processing technique that is used originally to detect straight lines within an image. This algorithm was utilized in [18] in order to find PRIs from a bit-map that was constructed from an interleaved pulse sequence. Plane Transform by [19] is another transform algorithm that converts the TOA sequence into two-dimensional bit-map, where one axis represents time and the other represents PRI. Both algorithms suffer from input PRI bandwidth limitations. The latter algorithm needs a mean in order to utilize its visual results.

The cumulative square sine-wave interpolation algorithm (CSSI) is utilized by [20] in order to extract periodic components of time of arrival differences. The advantage of this algorithm over the Difference-histogram, CDIF, SDIF, PRI-transform algorithms, and plane transform is its ability to sort jittered PRI accurately, compared to the others. However, the performance of the algorithm under a dense environment condition is unknown.

Clustering algorithms include the following: Self Organizing Feature Map (SOFM), Support Vector Clustering (SVC), Kernel Fuzzy C-Means (KFCM), Fuzzy-ART (F-ART), K-mean, and others.

There are four algorithms that use SOFM for de-interleaving. The first SOFM algorithm uses a validity index called Composed-Density-Between-and-Within-Clusters (CDBW) in order to find the right number of clusters [21]. The second algorithm modified the distance calculation method within SOFM in order to suit the radar environment, and in order to prevent degradation of the SOFM clustering ability due to agility in some radar features [22]. The last two algorithms use SOFM for clustering based on radar parameters that are extracted from wavelet features, and from resemblance coefficients [23] [24].

Four algorithm types were identified using Support Vector Clustering in the sorting of radar pulses. All of the proposed algorithms [25] [26] [27] [28] [29] [30] address the computational complexity of SVC. The algorithms are Cone Mapping Support Vector Clustering (CMSVC), Delaminating Coupling based on SVC (DCSVC), SVC and K-Means with Type-Entropy, and On-line Independent Support Vector Machines (OISVM).

Noone [31] proposed a modification to Fuzzy-ART in order to allow it to use some prior information regarding measurement accuracy. The use of prior information in Noone's algorithm does not mean, however, that the algorithm is supervised.

The F-ART algorithm can sort a high number of emitters simultaneously and with high right sorting rate [32]. Implementation on an application specific integrated circuit (ASIC) is done in [33]. Fuzzy-ART was compared with K-means clustering, Fuzzy-Min-Max clustering (FMMC), and Integrated Adaptive Fuzzy clustering (IAFC) in [33]. Fuzzy-ART was found to be the fastest, with the highest correct sorting rate using the available data set. Very-Large-scale Integration (VLSI) implementation for fast clustering is proposed in [34].

K-means is used by [35] for de-interleaving. The algorithm used AOA pre-sorting stage. In fact, the number of clusters (K) is unknown, which represents a serious problem for this algorithm.

There are various other methods such as Blind Source Separation (BSS) [36], Spectrum Atom Decomposition [37], Minimum Description Length [38], Radial Basis Function Neural Networks [39], and Immune Artificial Neural Network [40].

One algorithm combined the use of clustering with the time parameter. Chan [41] proposed clustering pulses into cells in three-dimensional space that consists of AOA, RF, and PW. Clustering is done based on Euclidean distance. The TOA parameter using a simple time-difference histogram is used to find the confidence level of each cell. It is unknown if this algorithm is able to prevent

cells from growing to form very big cells that contain pulses from different radars.

Chapter 4 The proposed de-interleaving algorithm is presented in this chapter. It is desired to have a de-interleaving solution that handles the agility in time, and at the same time, handles the agility in RF and PW. Time-based algorithms have problems with sequences that have time agility, while, clustering algorithms have problems with sequences that have agility in the rest of the pulse parameters (mainly RF and PW). The proposed solution, therefore, will use a time stage, along with a clustering stage.

The alternatives for using which of the clustering or the time stage before the other are discussed. This discussion moreover gives insight into the feasibility of using both the time stage and the clustering stage in the proposed solution. Time-based algorithms and clustering algorithms that were presented in the literature review chapter, which can be used in the time stage and clustering stage, were discussed.

It was found that none of existing clustering algorithms (with regard to the reviewed de-interleaving literature) satisfied all of the requirements of our desired solution. The desired clustering technique should, be unsupervised, it should not require prior information about clusters, it should be immune to noise, and it should have low complexity. It is therefore necessary to find an alternative clustering technique.

Density-based spatial clustering of applications with noise (DBSCAN) [42] is a clustering algorithm that has not been used before in the de-interleaving of radar pulses (to the best of our knowledge). However, as it satisfies all of our desired clustering features, it is thus used in the clustering stage of the system.

In the clustering stage, each cluster represents an emitter. However, the emitters with frequency agility will have multiple clusters. Therefore, these clusters are seen as different emitters in the clustering stage. The time stage is used to link the clusters of the same emitters. The timing of the clusters was utilized in testing the relationship between the different clusters.

The System Overview section presents the proposed de-interleaving system. The system is composed of different building blocks, which include Clustering, Pulse Sequence Segmentation, Agility Resolving, Continuous Cluster Sequence Detection, PRI Analysis, PDW filter, and Signal-to-noise-ratio (SNR) Filter. The DBSCAN algorithm is used in the clustering stage of the system. The general concept of DBSCAN is illustrated in Section (4.4.1.1). The flow diagram of the applied clustering algorithm is presented. Pulse Sequence Segmentation plays an important role in the management of the input buffer. This block contributes to the enhancement of algorithm performance (speed), as described in Section (4.4.2). Thereafter, the Agility Resolving Block is presented

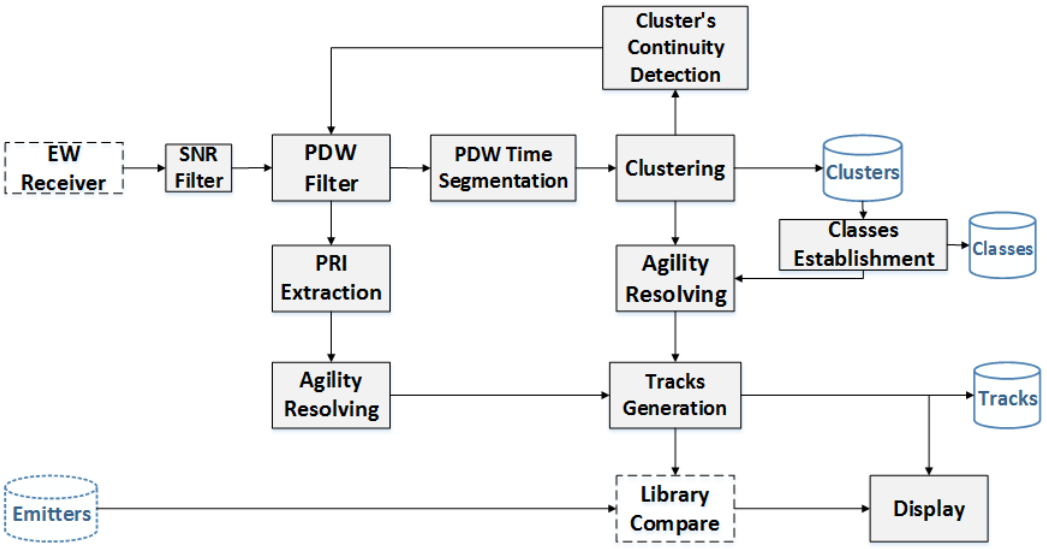


Figure 1.3: Proposed de-interleaving system.

in Section (4.4.3). This block consists of two sections: Agility Hypothesis Generation, and Agility Hypothesis Test. In fact, the time parameter is utilized for resolving of the agility. Some related statistical analysis is provided within this section. The Cluster's Continuity Detection block is presented in Section (4.4.4). This block assists in preventing radars that have a continuous pulse stream at the input of the EW receiver from causing problems in the agility resolving stage. Moreover, this block contributes to the performance (speed) of the algorithm. The PRI Analysis block is intended for clusters that have continuous pulse streams. This block is presented in Section (4.4.5). PDW Filter is presented in Section (4.4.6). This filter removes PDWs from the input buffer according to the feedback received from other blocks of the de-interleaving algorithm. Finally, the SNR Filter is presented in Section (4.4.7). This filter rejects PDWs that have a poor SNR.

Chapter 5 This chapter present the test results of the proposed de-interleaving algorithm. The chapter consists of three sections. The results in the first section were used to support the decision to use DBSCAN in the clustering stage of the de-interleaving algorithm, in addition to supporting some other design decisions.

The test of DBSCAN mainly focuses on the ability of the clustering algorithm to cluster radar pulses in the presence of outliers and measurement errors of the EW receiver. The test shows the high accuracy of clustering using DBSCAN under these conditions and under different environment density scenarios.

The clustering algorithm was tested against data that contained a number of clusters, up to 60 clusters. It is assumed that all clusters are formed under the condition of simultaneous reception

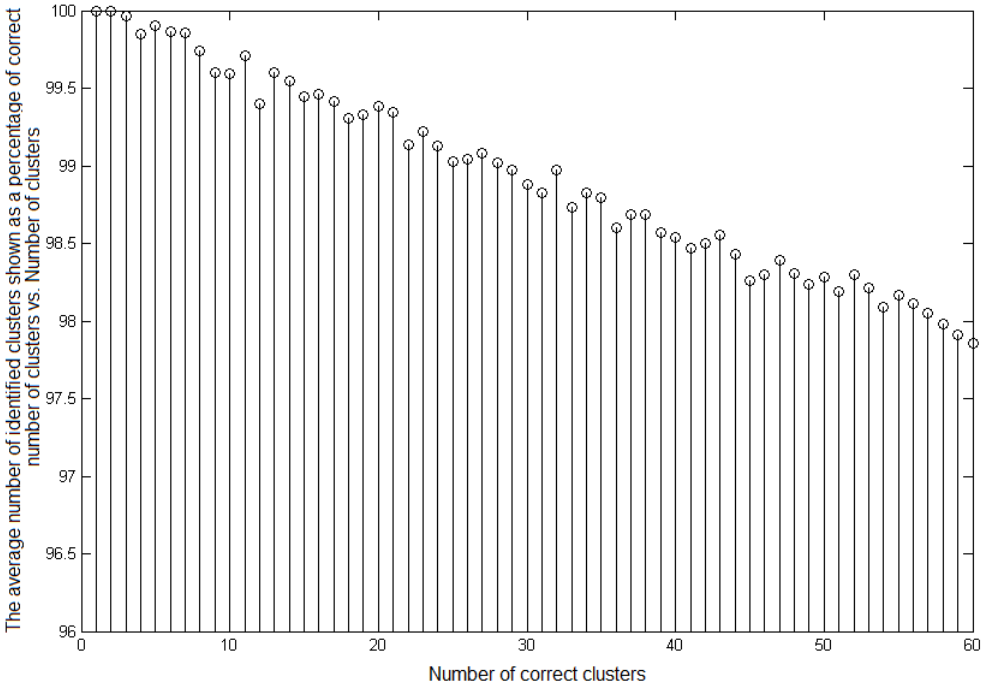


Figure 1.4: The average number of identified clusters (shown as a percentage of the number of identified clusters) vs. the number of correct clusters. Note that the y-axis is shown between 96 and 100. The data contains noise (outlier) pulses with a ratio of 5 %.

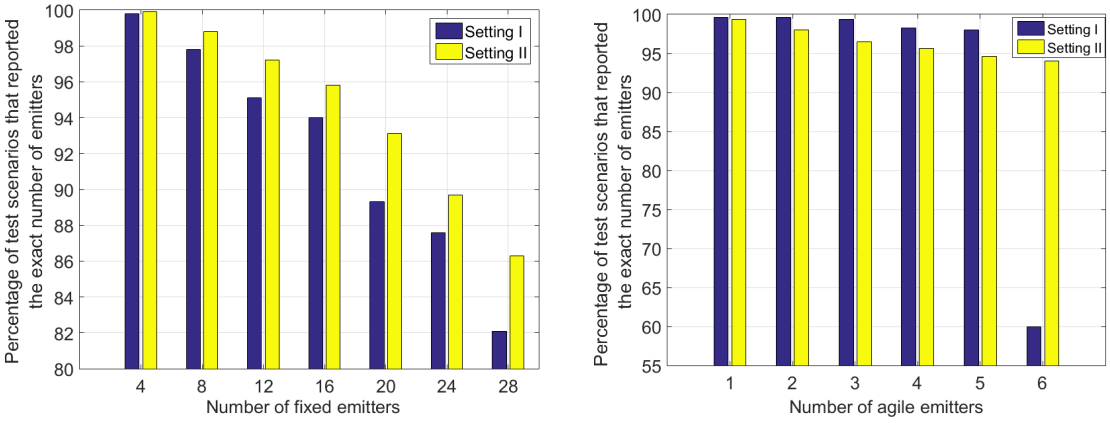
from the various emitters (all emitters are pointing towards the EW receiver at the same time). The used parameters are AOA and RF.

The AOA centers of the clusters are uniformly distributed, and so are the RF centers. Each cluster has a Gaussian error distribution. The RMS errors of the measured RF parameters and AOA parameters are 1 MHz, and 5^o, respectively. Each cluster, moreover, has a random number of pulses, which are uniformly distributed between 5 and 50.

The simulation was repeated 1000 times for each of the test scenarios (where each scenario has a different number of clusters). The simulation results are shown in Figure (1.4). The figure shows that the correct clustering rate is very high. For example, the correct clustering rate with regard to 30 clusters is 98.88%. This corresponds to an average correct clustering of 29.7 clusters out of 30. In other words, the probability of correctly identifying all of the 30 clusters is very close to 100%.

In the second section of Chapter 5, the proposed de-interleaving algorithm is tested against three test scenarios. The first scenario is for environments, where all emitters are fixed frequency emitters; the second scenario is for environments, where all emitters are frequency agile; and the third scenario is for environments, where some emitters are fixed while others are agile.Three criteria were used to present the results.

The first test criterion requires the algorithm to report all of the received emitters correctly,



(a) Fixed frequency emitters.

(b) Agile frequency emitters

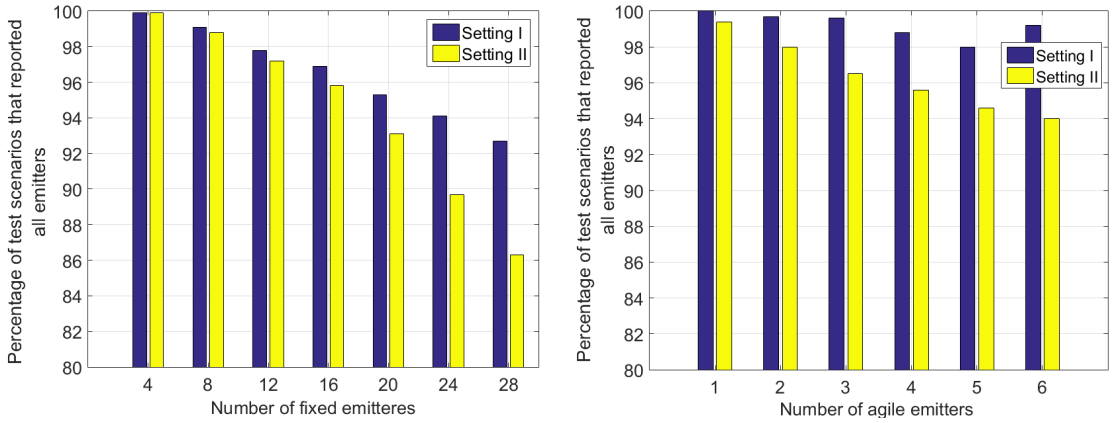
Figure 1.5: Percentage of tests that have reported the exact number of emitters (the first test criterion).

without reporting any false emitter. The algorithm can use two settings, viz., Setting (I), and Setting (II). The test for fixed frequency emitters shows that the algorithm has better performance in Setting (II) compared to Setting (I). Setting (II) only reports an emitter if it has at least two clusters formed at two different instances of time. This will reduce the chance of incorrectly generating additional clusters within the algorithm, which could happen due to the segmentation stage. Setting (I) requires at least one cluster for a given emitter in order to report that emitter. Therefore, the algorithm under Setting (I) is more susceptible to incorrectly generated clusters due to the segmentation stage.

The worst case for accuracy based on the first criterion was 82% using Setting (I) and 86% using Setting (II) for 28 emitters. However 99% accuracy is achievable for a lower number of emitters as shown for the case of four emitters. Figure (1.5) shows the test results for the first criterion.

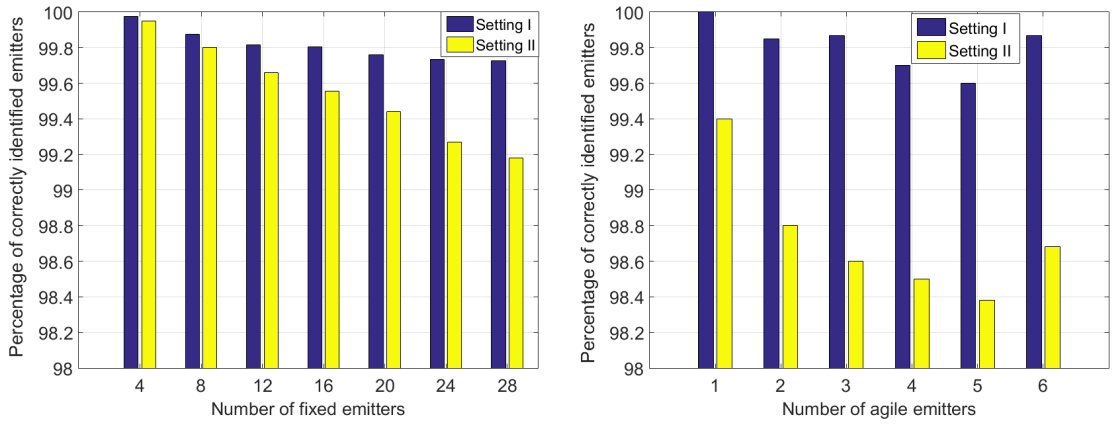
The test of the agile emitters showed a better results (performance) than those of the fixed emitters, when comparing the results with respect to the number of utilized clusters in the test. The test consists of up to 24 clusters, which could be incorrectly reported as 24 emitters (in the agile frequency test), making more challenge on the agility resolving stage. The number of overall clusters in this test is almost similar to the fixed frequency test; however, the number of emitters is lower. The high accuracy achieved compared to the previous test is an indication of the effectiveness of the agility resolving stage. The accuracy (using the first test criterion) will drop significantly if this stage was not effectively resolving frequency agility.

The second criterion is less conservative than the first criterion, in that it tolerates incorrectly reported emitters, if all the received emitters are reported correctly. Setting (I) shows better results



(a) Fixed frequency emitters. (b) Agile frequency emitters.

Figure 1.6: Percentage of tests that have reported all of the emitters (the second test criterion).



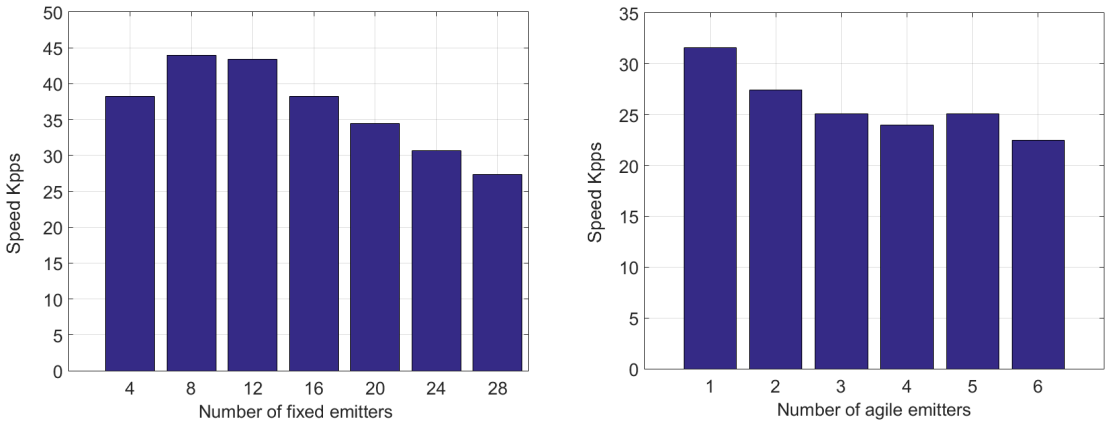
(a) Fixed frequency emitters. (b) Agile frequency emitters.

Figure 1.7: Identification ratio of emitters (the third test criterion).

compared with Setting (II) as expected. The accuracy using this criterion was better than 86%, and up to 99%.

The third criterion is the ratio of the correctly reported emitters across all the tests. In other words, it gives indication of the chance of a received emitter being reported by the algorithm. The test shows that it is better than 99% in the case of fixed frequency emitters, and better than 98% in the case of agile frequency emitters. Setting (I) was slightly better than Setting (II) for both fixed and agile frequency emitters.

In fact, the three criteria together give a good picture about the effectiveness of the algorithm. The accuracy of the algorithm in general degrades as the number of clusters increases. However, the algorithm showed very good de-interleaving accuracy results over the tested ranges of emitters for the different scenarios.



(a) Fixed frequency emitters. (b) Agile frequency emitters.

Figure 1.8: Average speed of the algorithm.

The average speed of the algorithm under different test scenarios was reported in Figure (1.8). This gives an indication about the capacity of the algorithm when handling emitters with a high pulse rate.

In comparison to similar algorithms, the algorithm provided by Zhifu [35], for instance, used k-means clustering in its de-interleaving solution. The de-interleaving algorithm was tested against twelve fixed frequency emitters. The correct sorting rate was 99.64% according to [35]. The test result for equivalent number of clusters (12 clusters) for DBSCAN was 99.40%. However, the test performed for DBSCAN was extremely strict by assuming that, all of the emitters in the test were pointing toward the EW receiver in the same time. Better clustering results are achieved in the normal EW environment conditions, in other words, a 99.82% correct clustering rate, as can be seen in Figure (1.4).

DBSCAN was tested for 60 emitters (in contrast to only 12 emitters for k-means in [35]) with a very high correct clustering rate. The test of DBSCAN (provided in the dissertation) was performed by using 1000 independent run. In contrast, the results in [35] (which utilized k-means algorithm) appear to be based on single test. This is important to be mentioned because the algorithm provided in [35] can be very dependent on the test data. Moreover, the test performed for DBSCAN was extremely strict in that it assumed that all of the emitters in the test were pointing toward the EW receiver at the same time. Better clustering results can be achieved in the normal EW environment conditions as can be seen in Figure (5.6). The algorithm proposed in this dissertation has a correct clustering rate of more than 99.8% for 12 fixed frequency emitters compared to the correct clustering rate of 99.64% in Zhifu's algorithm.

The proposed de-interleaving in this dissertation deals with agility in frequency, in contrast

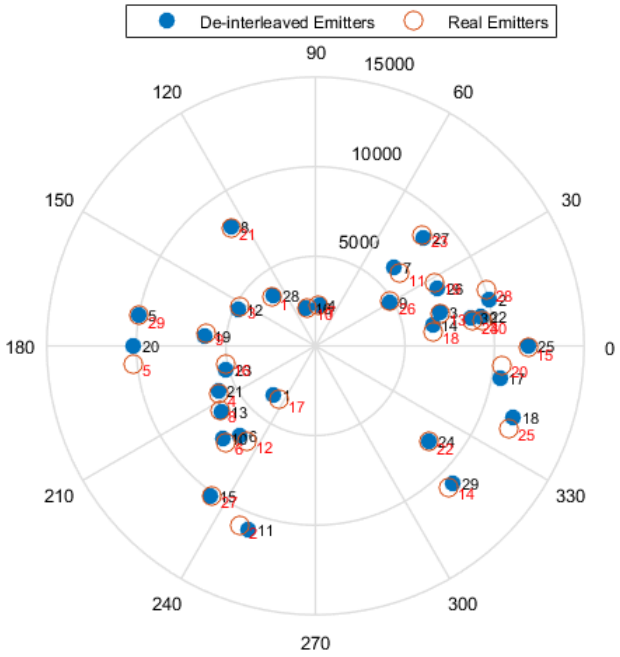
with Zhifu's algorithm which does not provide a solution for agile frequency emitters. This can be seen clearly from the second stage of Zhifu's algorithm which uses the RF parameter with the PW parameter in the clustering. Therefore, agility in RF becomes problematic for Zhifu's algorithm.

In this dissertation, therefore, the proposed algorithm provided a solution for resolving frequency agile emitters. The results of de-interleaving up to 6 radars each of which has four frequencies are presented in Figure (5.10), while the rate of successfully de-interleaving emitters is better than 99.6 %.

The proposed algorithm avoided utilizing the PW parameter in de-interleaving, and therefore PW agility is not a problem for the proposed algorithm. Hence, it was not necessary to perform the test in conditions of PW agility.

In the same sense, the PRI (or ∇TOA) parameter was not utilized in de-interleaving, but instead the time of cluster was used. The time of cluster is not sensitive to the agility of the PRI parameter, and therefore the test was not performed under this condition. Furthermore, the clustering stage does not even utilize any kind of time parameters.

In the third section of Chapter 5, test samples for the proposed de-interleaving algorithm are given. The samples show a presentation of the interleaved PDWs. The results were compared with the simulated threat library, and the comparison results are visualized. The samples also present the Emitters-Tracks table, which is generated by the de-interleaving algorithm. This table contains descriptions about each de-interleaved emitter, and these descriptions include information about agility, frequency, direction, and various other descriptions.



(a) Real emitters vs. de-interleaved emitters for a test sample. The numbers appear in the plot are emitters' IDs.

Fields	ID	Library_ID	clustersCount	ClssesIDs	RF_center	RFagility	RFagili	RF_hops	AOA_	Tracking	sim	sir	confirmed
1	1	6	5	1	9605	'Fixed'	[]	9605	115	[]	[]	[]	1
2	2	2	19	[2,3,4,5]	2990	'Agile'	4	[2691,2890,3091,3288]	-25	[]	[]	[]	1
3	3	3	5	6	3402	'Fixed'	[]	3402	-70	[]	[]	[]	1
4	4	4	5	7	11072	'Fixed'	[]	11072	-40	[]	[]	[]	1
5	5	9	5	8	5798	'Fixed'	[]	5798	-35	[]	[]	[]	1
6	6	7	5	9	11291	'Fixed'	[]	11291	-100	[]	[]	[]	1
7	7	8	5	10	11689	'Fixed'	[]	11689	130	[]	[]	[]	1
8	8	1	18	[11,12,13,14]	7449	'Agile'	4	[6703,7201,7697,8194]	-110	[]	[]	[]	1
9	9	5	5	15	7214	'Fixed'	[]	7214	120	[]	[]	[]	1
10	12	10	1	19	10206	'Fixed'	[]	10206	110	'Tracking'	[]	[]	1

(b) Emitters' Tracks-Table generated by de-interleaving algorithm, for another test sample.

Figure 1.9: Samples from demonstration test examples.

Chapter 6 This chapter presents the conclusion of the dissertation, which provided a de-interleaving solution for on-line ELINT application, particularly for ELINT/ESM receivers. The solution takes into consideration the realistic EW hardware and the EW environment.

The dissertation contributed to this field by utilizing the DBSCAN clustering algorithm, which has not been used before in the problem of de-interleaving, and it proved that, this algorithm is indeed effective in handling the de-interleaving problem. The provided de-interleaving solution is capable of handling agility in time, as well as agility in frequency and PW. At the same time, it is a feasible solution, which does not require complex integration with an EW receiver, nor does it interfere with the signal processing unit of the EW receiver. The dissertation also discussed on-line ELINT applications that have not got attention in the literature.

The dissertation suggested specifications for a practical and feasible system, taking in account

real EW systems. The importance of this work lies in supporting future research, by providing the researchers with a literature survey that facilitates selection of design decision, once different requirements are proposed. It also provides a brief and relevant introduction to EW for the purpose of on-line ELINT de-interleaving.

The proposed algorithm was found to be effective in de-interleaving both fixed and frequency agile emitters. At the same time, the algorithm is able to deal with time agility. It is also capable of working in a dense emitter environment. The accuracy and speed of the algorithm were proven to meet the requirements provided, with a window for further enhancement of speed. The de-interleaving algorithms available in the literature were found to be unsatisfactory in terms of meeting the requirements of the recommended de-interleaving system, as described in Chapter 1.

The solution does however have a limitation with regard to the number of emitters, in that it should not handle more than 30 simultaneous emitters (or clusters) in order to provide the best performance. However, even in the case of more than 30 simultaneous emitters, the performance does improve after a short time of running the de-interleaving algorithm. The algorithm can thus work with more than 30 emitters, with some compromise in performance.

The future work of the dissertation includes taking the movement of the EW receiver into consideration. It also, includes enhancing the optimizing the implementation of the algorithm in order to enhance the average speed of the algorithm by considering different types of implementation. Finally, the future work should increase the capacity of the algorithm to handle more number of clusters without compromising the current speed and performance of the algorithm.

In Chapter 1, it was assumed that, all emitters are stationary. In the case of moving emitters, the algorithm should be functional under some circumstances, but it was not designed or tested for this purpose.

Future work thus includes adding the capacity to deal with moving emitters within the algorithm. It also includes further enhancement to the speed of the algorithm.

1.9 Summary

In this chapter, the topic of the research was introduced. The chapter started by presenting the background to illustrate the need for a de-interleaving algorithm, and to explain in which applications it can be used. In particular, it illustrated the need for de-interleaving in online ELINT applications. The aim of the dissertation, as stated in this chapter, is to propose an appropriate de-interleaving algorithm for use with the ELINT/ESM receiver (with online application). The chapter also presented the objectives, the approach, the assumptions, the scope, and the motiva-

tion of the dissertation. Thereafter, some related background was presented to define and explain some of the important terms that had been mentioned in the Introduction. Finally, the chapter-by-chapter organization of the dissertation was introduced. Chapter 2 will focus on the background of the commonly measured radar parameters, which are used in the de-interleaving algorithm. It will also present the background of the EW receiver that is used in the measurement process.

Chapter 2

Theoretical Background

De-interleaving algorithms process radar parameters that are measured by an EW receiver. Therefore, it is important to have a background understanding of these parameters, and about the system that measures them. This background is important before proceeding to the literature review. In addition, it is important in order to assist us in making the right design decisions regarding the proposed de-interleaving solution.

This chapter thus provides a background about the most common pulse parameters that are measured by EW receivers. It also provides a brief background about the particular EW receiver of interest, which is an ELINT/ESM receiver (the wideband digital receiver).

2.1 Measured radar parameters

Many radar parameters can be measured in EW. This section discusses the commonly measured parameters, especially those related to de-interleaving. The pulse parameters that are most commonly measured by EW receivers are TOA, AOA, RF, PW, and PA. All these parameters can be obtained from a single pulse. PRI is a very important and common parameter. In some EW receivers, it is necessary to do de-interleaving in order to obtain this parameter, whereas in other EW receivers de-interleaving is done based on PRI. Another pulse parameter that becomes common is the modulation type. Polarization is one of the radar parameters that can be measured, but it is rarely found in EW receivers, due to unnecessary hardware complexity.

Some other parameters can only be obtained after de-interleaving, such as ASP and HPBW. These parameters can be used for classification or identification. A pulse-rise-time parameter can be obtained (theoretically) from single pulse. This parameter is used for purposes of identification (also called Specific Emitter Identification (SEI)) and needs special measurement conditions.

Therefore, measuring the pulse-rise-time parameter and some other parameters is not practical before completing the de-interleaving process.

The common parameters are briefly discussed in the following subsections.

2.1.1 Pulse Amplitude (PA)

EW receivers usually measure the average amplitude value for a given pulse. PA is obtained from the envelops of the measured pulse. It can experience some fluctuation due to noise, multipath, or even due to the modulation type used. Moreover, PA changes from pulse to pulse in case of antenna scanning. PA is not regarded as a useful de-interleaving parameter in literature. Current EW receivers can have an amplitude dynamic range of 60 dB or better.

2.1.2 Time of Arrival (TOA)

TOA is the time that the pulse arrives to the EW receiver, measured with reference to the EW receiver's clock. The rising edge of the pulse determines the TOA. The PRI parameter is derived from the TOA of two consecutive radar pulses. The PRI parameter will be discussed in a separate section (see Section 2.1.3). For high SNR, the accuracy of the measured TOA with a fixed amplitude threshold is given by [9];

$$\sigma_{TOA} = \frac{1.25\, t_R}{\sqrt{2\, SNR}} \tag{2.1}$$

where σ_{TOA} is the Root Mean Square (RMS) error of the measured TOA due to noise, t_R is the duration between 10% and 90% points of rising edge, and SNR is the signal to noise ratio of the receiver.

2.1.3 Pulse Repetition Interval (PRI)

PRI is the time between the leading edges of two consecutive pulses. PRI parameter can be fixed or variable, although most of radars have fixed PRI [43]. There are many types of PRI patterns. The PRI used will strongly affect the performance of the radar.

PRI is given by;

$$PRI_{ij} = TOA_i - TOA_j \tag{2.2}$$

Based on equation (2.1), the accuracy of PRI measurement is [9]:

$$\sigma_{PRI}^2 = \sigma_{TOA_i}^2 - \sigma_{TOA_j}^2 \tag{2.3}$$

hence;

$$\sigma_{PRI} = \frac{t_R}{0.8\sqrt{SNR}} \tag{2.4}$$

The common PRI patterns are: Fixed PRI, Jittered PRI, Staggered PRI, Dwell and Switch PRI, Sliding PRI, Scheduled PRI, Sinusoidal PRI, and Group PRI [9], which explained below.

Fixed PRI The value of the PRI is constant. The variation from the mean value, which is an unintentional variation, is less than 1% [9].

Jittered PRI The value of the PRI from one PRI interval to the next is intentionally changed in a random manner. PRI (or time) jitter is an EP technique that is used against false targets deception jamming [44]. The variation can be as large as 30% from the mean value [9]. Time jitter is not desired in radar because it degrades the performance of the Moving Target Indicator (MTI) and imposes limitations on the improvement factor of pulse compression [45]. There are two types of jitter, viz., cumulative, and non-cumulative. In the ideal case when no jitter exists, the values of TOA are the integer multiples of the PRI (assuming the first pulse is the reference). In the case of non-cumulative jitter, the drift of the current TOA value from the TOA of the ideal case is only dependent on the random error that is added to the current TOA. In the case of cumulative jitter, the drift of the current TOA from the TOA of the ideal case depends on the accumulation of all added random errors of all previous pulses.

Staggered PRI In staggered PRI, different successive PRI values are used. The PRI intervals used are called stagger positions and are sometimes called stagger levels. Staggered PRI has a number of positions (N) that is greater than one. Adding N successive PRI values will form what is called a staggered frame. The staggered frame interval is periodic in converse to the PRI positions that construct the frame. Furthermore, combining N successive PRI intervals will form a periodic interval, regardless of the starting position. Therefore, staggered PRI can be seen as a combination of various fixed PRIs, all of an interval value equal to the stagger-frame interval. Staggered PRI is used in order to overcome the problem of blind speed in search radars that use MTI [46].

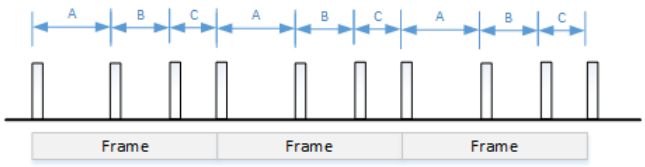


Figure 2.1: Illustration of Staggered PRI. The figure illustrates an example of three levels of staggered PRI.

Dwell and Switch PRI Many different fixed PRIs are used in this pattern. The radar performs fast switching between these fixed PRIs [9]. The time duration that is spent by the radar transmitting a given PRI before switching to the other is called dwell time. This PRI pattern is used in resolving range ambiguity, velocity ambiguity, and blind range (eclipsing) of radar. Moreover, this PRI pattern can be used as an EP technique. The switching between PRIs is performed automatically by the radar according to its desired function.

Sliding PRI In this pattern, a continuous increase or decrease in PRI value is performed. This pattern is used to reduce the unambiguous range of a radar, as it searches a higher elevation, in addition to being used in eliminating blind range [9].

Scheduled PRI This PRI pattern is used by track-while scan radars. The transmitted PRI is dependent on the receiver location and on the number of tracked targets [9].

Sinusoidal PRI This pattern is mostly used with conical scanning tracking radars. The frequency of PRI modulation is usually close to 50 Hz and with a 5% modulation index [9].

2.1.4 Radio Frequency Parameter (RF)

The frequency parameter of interest in PDW is the mean frequency value within the pulse [43]. In frequency agile radars, the radar changes its frequency on either a pulse-to-pulse basis or a burst-to-burst basis [2]. A radar with a time-frequency-coded waveform is an example of pulse-to-pulse frequency agility. The radar that uses this type of waveform transmits a group (burst) of N pulses, each pulse with a different frequency [45]. For radars that use Doppler processing or coherent processing, the frequency does not change during processing interval [9]. The processing interval usually includes several pulses, and hence, the frequency changes on a burst-to-burst basis. The agility band of radars is mostly limited to 10% of the center frequency [3]. Frequency agility is used as EP technique against noise and deception jamming, in addition to being used in radar performance enhancement.

Different techniques are utilized in EW receivers for frequency measurement. The accuracy of the measurement is dependent on the technique used. An Instantaneous Frequency Measurement (IFM) receiver can cover huge instantaneous bandwidth (2-18 GHz) with accuracy in the range of 5 MHz. IFM can provide high Probability-of-Intercept (POI), but it can not measure simultaneous signals. Fast Fourier Transform (FFT) and Digital IFM (DIFM) are other possible methods, where the latter is the digital version of IFM that is applied on IF band of a superheterodyne receiver. The accuracy of frequency measurement using the IFM technique is given by [9] as follows;

$$\sigma_f\,(rms) = B_V \sqrt{\frac{1}{3\,SNR}\frac{B_V}{B_R}} \tag{2.5}$$

where: B_R, B_V are input bandwidth and output bandwidth of IFM respectively in Hertz. SNR is the input signal-to-noise ratio.

Knowing that B_V is inversely proportional to PW, one can note from equation (2.5) that, the accuracy enhances with longer PW and with better SNR. Generally, for digital wideband EW receivers, 1 MHz accuracy is expected.

2.1.5 Pulse Width (PW)

Pulse width is the time duration between the rising edge and the trailing edge of a pulse envelope. The minimum measured PW is usually $100\mu s$ [43]. The maximum PW is determined by the designer of the EW receiver. Any PW exceeding the maximum PW can be regarded as a Continuous Wave (CW). The accuracy of PW measurement can be affected under multipath propagation condition, and PW is thus not regarded as a reliable parameter for de-interleaving.

2.1.6 Angle of Arrival (AOA)

AOA is a very useful radar parameter for de-interleaving [43]. This because the AOA of a radar does not change rapidly from pulse to pulse, even under very high target speed. AOA should be measured on a pulse-by-pulse basis in an EW receiver in order to be useful as a de-interleaving parameter. Two AOA parameters can be measured for radar, viz, Azimuth angle and Elevation angle. Most EW receivers only measure Azimuth angle. The subsystem that is used in AOA measurement is called Direction Finder (DF). There are many DF techniques, and the accuracy of AOA measurement is dependent on the the particular technique used. The designer of EW system uses the technique that best suits the desired functionality of EW receiver and fits the constraints. Generally, the common DF methods are amplitude comparison, phase interferometry, time-difference-of-arrival (TDOA), and narrow beam search antenna [47]. The amplitude and phase

are the most popular DF techniques in EW. The amplitude method can cover a wider frequency band, but with a compromise in accuracy. The phase method has better accuracy but can work only on a narrow band. Sometimes both methods are combined to achieve higher bandwidth coverage and good accuracy at the same time. In fact, EW receivers have different spatial search strategies. The receiver can have instantaneous isotropic angular coverage, or it can switch between a small number of instantaneous coverage sectors.

A narrow band search antenna is very useful in specific situations, such as when POI is not important. However, narrow-beam antenna search methods do not always measure AOA on pulse-by-pulse basis. In fact, in spite of the usefulness of narrow beam search methods in some ELINT scenarios, it is still not a popular technique in EW, as Tsui notes in [47]. TDOA is an accurate DF method, but it needs large spatial separation between antennas, hence it is not suitable for EW receivers with limited size constraints. There are also other less popular methods such as, Doppler shift method, beam-forming methods, and microwave lens.

The accuracy of AOA measurement is dependent on many factors, such as the SNR and the configuration of DF. Generally speaking, the accuracy of AOA in an ELINT/ESM receiver is in the range from 1^o to 5^o.

2.1.7 Modulation on Pulse (MOP)

Radars may have intra-pulse modulation for different reasons, such as enhancing range or Doppler resolution, enhancing detection, etc. PDW generated by the EW receiver can include field called Modulation-ON-Pulse (MOP). This field could indicate the presence of intra-pulse modulation or even the type used. The important modulations types are: Linear Frequency Modulation (LFM), Nonlinear Frequency Modulation (NLFM), and Phase Coding [36]. Common examples of phase coding are Barker coding, and Frank coding. The MOP can be represented within PDW by a flag in order to indicate whether the modulation exists within the pulse or not. The MOP parameter might also be represented by the following values: “LFM”, “NLFM”, “Parker”, and “Frank”.

2.2 ELINT/ESM Receiver

De-interleaving is one of the tasks performed by an EW receiver. Therefore, it is important to understand how EW receivers work in general terms, before proceeding with a discussion of de-interleaving. The wideband digital receiver is the most appropriate receiver for ELINT/ESM systems. In this section, a brief but relevant introduction to EW receivers is thus presented.

An EW receiver covers a very wideband, usually from 1to 18 GHz. In fact, some systems cover from 0.5 GHz up to 40 GHz. When covering such a wide band, the system usually consists of different sections: One section covers the band from 0.5 MHz to 2 GHz, another section covers the band from 2 GHz to 18 GHz, and another section covers the band from 26 GHz to 40 GHz. The receiver's instantaneous band width (IBW) is at least 500 MHz. Some receivers can reach more than 1000 MHz in order to cover the agility of some radars, and in order to increase POI. EW system can have different innovative configurations. Usually two types of receivers are utilized within an ELINT/ESM system. One receiver is called IFM, which covers huge IBW usually from 2 to 18 GHz. This receiver usually measures only frequency. IFM can measure short pulses with a resolution finer than FFT based receivers [48], but it can not measure simultaneous signals. The other receiver is the superheterodyne, which can measure all parameters, including frequency, with better accuracy and better sensitivity. This receiver has wide IBW (1 GHz), but this IBW still very small compared to the EW frequency coverage range (more than 17 GHz). Channelization is commonly used in EW receivers in order to enhance their sensitivity. In addition, it allows for handling of simultaneous pulses.

A possible EW receiver configuration is illustrated in Figure (2.2). De-interleaving is performed within the Data Processing Unit (DPU). As shown in the diagram, the input to DPU is PDW. In ESM receivers, the latency is a critical factor, and hence, the structure shown in Figure (2.2) can be optimized to reduce such latency. This can be done by combining the DPU with the Digital Signal Processing (DSP) unit. The threat library can also reside on the same unit. ELINT does not have a critical latency requirement and hence the configuration in Figure (2.2) is an appropriate one. Because DSP unit resources are limited compared to the required signal processing tasks, performing de-interleaving on a separate unit (as in Figure (2.2)) will provide more flexibility in an ELINT situation.

The antenna subsystem usually consists of an array of antennas that are required for the DF function. A superheterodyne receiver needs one antenna that can be shared with DF. An omni-directional antenna is usually used (in addition to an antenna array). This omni-directional antenna is used for IFM in order to provide high POI.

In another configuration, the antenna subsystem could consist of directive narrow-beam antenna. This antenna is rotating mechanically in order to scan the azimuth dimension. Two directive antennas (instead of one) with a small squint are used when a mono-pulse concept is used for DF. The omni-directional antenna in this case is used by DF in order to discriminate between main-lobe and side-lobe intercepted signals. POI for this configuration is very low and not suitable for ESM.

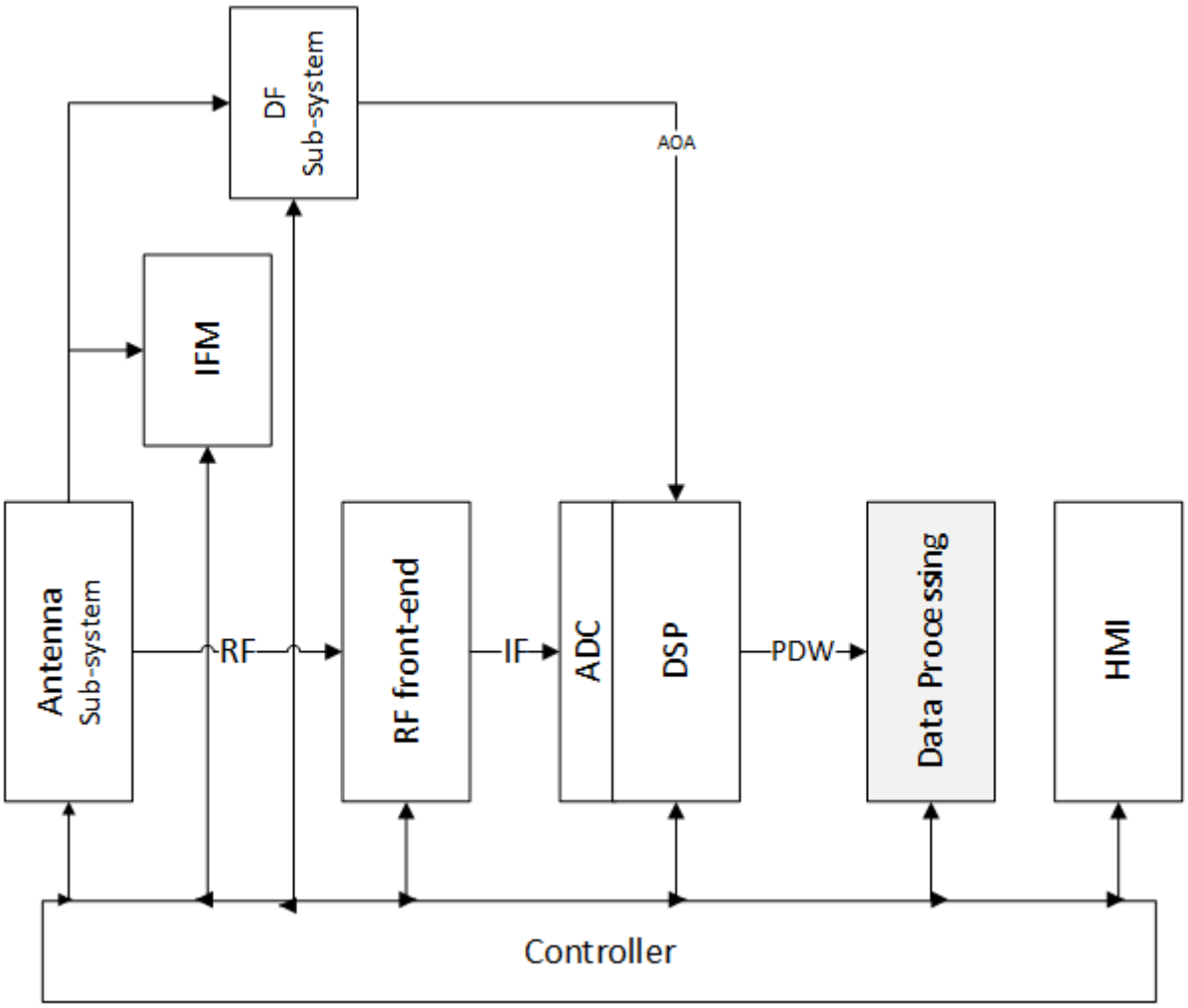


Figure 2.2: Possible EW digital receiver configuration.

A superheterodyne receiver has an IBW that is very small compared to the full frequency range. It is possible to use multiple superheterodyne receivers in order to cover the full frequency range, but this solution is impractical due to its large size, high cost, and large power consumption.

IFM is an analog RF receiver that measures frequency, but this receiver cannot measure frequency in the presence of simultaneous signals. The advantage of using this receiver is the huge instantaneous bandwidth. Therefore, when IFM is connected to omni-directional antenna, it provides high POI.

If the receiver is in ELINT mode, it can scan both the frequency range and the azimuth spatial dimension. IFM can be used in order to optimize the search strategy. The spatial dimension is divided into a number of sectors, depending on the antenna array structure that is used. If full spatial coverage is required, then an omni-directional antenna is used instead of an antenna array.

The signal received by the antenna passes to the RF front-end. The frequency is down converted to the Intermediate Frequency (IF) by mixing with the Local Oscillator (LO) frequency, which is provided by a high-speed synthesizer. The bandwidth of IF is wide (possibly 1000 MHz), therefore a filter bank with lower bandwidths is usually used in the RF front end, in order to allow for compromising IBW to achieve better SNR in some situations.

The IF signal passes to the DSP unit after being digitized with the Analog-to-Digital-Converter (ADC) at a sufficient sampling rate. The sampling rate is usually in the order of Giga samples per second, which is a high data rate that needs parallel processing by mean of FPGA or ASIC. The tasks that the DSP unit performs include generation of an In-phase and Quadrature (I/Q) signal, detection, decimation, FFT, parameter extraction, and others. The IBW is high and therefore digital channelization is usually used to enhance both sensitivity and SNR. Channelization furthermore reduces the problem of measuring simultaneous pulses. In addition to the need for high speed processing, these tasks require plenty of FPGA resources. The output of the DSP unit is PDW.

De-interleaving, classification and identification are done within the DPU. The input to this unit is the PDW. The de-interleaving results are shown to an operator in the form of radar tracks. Radar tracks are compared with a threat library. If a certain radar track matches an entry in the threat library, then the radar name and symbol are displayed. If no matches for the track were found, then “Unknown” is attached to the track. The operator decide whether to add unknown emitter tracks to the threat library. The interaction between the operator and the EW system is obviously done through a Human Machine Interface (HMI).

2.3 Conclusion

The commonly measured radar pulse parameters of EW receivers include TOA, AOA, RF, PW and PA. Amongst these five parameters, TOA, AOA and RF are the most reliable parameters for use with de-interleaving. All the parameters mentioned above are obtained from a single pulse. Some parameters could add more complexity to the receiver, which would make their use uncommon in EW receivers; these are, for instance, the polarization parameter and some of the intra-pulse modulation parameters. In addition to the parameters above, certain parameters need de-interleaving a priori, i.e. beforehand, such as ASP. Most of these parameters are used for the purpose of classification or identification, but not for de-interleaving.

An EW receiver can have different configurations, and may utilize various technologies and techniques in the measurement process. As a result, the accuracy and performance of different EW receivers are different, and this needs to be taken into consideration when proposing a de-interleaving solution.

In this chapter, a brief theoretical background was presented about the measured radar parameters, in addition to the particular EW receiver of interest, viz. the ELINT/ESM receiver. In Chapter 3, the literature is reviewed, specifically with regard to the algorithms that are related to the de-interleaving of radar pulses, and in particular to the on-line ELINT application.

Chapter 3

Literature Review

This chapter reviews the published algorithms regarding de-interleaving of radar pulses. The algorithms, in this review, are classified into three groups: time-based algorithms, clustering algorithms, and combined algorithms.

3.1 Time-based De-interleaving Algorithms

Time-based algorithms rely primarily on the time of arrival (TOA) parameter in the de-interleaving of radar pulses. This section discusses different algorithms, which include the following: Sequence Search (SS), Time Difference Histogram (TDH), Cumulative Difference Histogram (CDIF), Sequential Difference Histogram (SDIF), PRI transformation, Time-Period PRI Transform, Hough Transform, Plane Transform, Cumulative square sine-wave interpolation, Active Cell TOA De-interleaving, Time Progressive Processing, Multiple Correlation Hypothesis, Pandu Algorithm, Pulses Auto-Correlation, and Folding algorithm.

3.1.1 Sequence Search

Generally, a radar pulse sequence can be extracted from an interleaved sequence if its PRI and and its PRI phase are known. The PRI phase here is the time displacement from an arbitrary selected time reference. For a pulse sequence of a single emitter, the measurement of PRI can be performed by finding the time difference between a pair of consecutive pulses. Obviously, this is not true for an interleaved sequence. For an interleaved pulse sequence, there are plenty of possible combinations of pulses, but only a few of them form correct pulse-pairs. A sequence search algorithm searches for these correct pulse-pairs among a large number of pairs.

In brief, a pulse-pair is randomly selected from an interleaved sequence, and assumed to be a correct pair. In order to test if this assumption is true, the algorithm will then search for pulses at time locations that are integer multiplications of the assumed PRI. The search in time will begin from the selected pulse pair. The PRI value is confirmed to be correct if the search is successful. A successful search should satisfy a predefined condition, such as, the presence of pulses at three consecutive search locations (at least). Projection in time, both in forward, and in backward directions, will be performed in order to extract the whole sequence of the confirmed PRI. If the sequence search failed to confirm the PRI under test, then another random pulse pair is selected. This process will continue until the remaining pulses are less than a predefined number.

When the searched sequence is time jittered, a tolerance in the search is required in order to avoid missing a true pulse. The tolerance of the search is thus set according to the expected jitter ratio. The tolerance window need to be wide enough to include more than one pulse (from different radars), which may confuse the search process. A similarity test was proposed in [6] in order to overcome this problem. Euclidean distance is used as a mean to measure the similarity between pulses, using parameters such as AOA, RF and PW. Therefore, the similarity test will be applied for each searched pulse. If a targeted pulse was found to be within some similarity (distance) threshold, then the pulse would pass into extracted sequence. The weight of each parameter in Euclidean distance is tuned in order to give importance to some parameters over others. The sequence search concept presented by [7] is not very different from [6].

In general, the sequence search is not efficient for a sequence that consists of many PRIs, because sequence of N pulses could have computations in the order of $N^3/3$ [11].

Recurrent Neural Network (RNN) that is based on the state space model of a radar sequence was proposed by [8] in order to estimate and track radar sequences. Noon [49] combined Sequence-Search with RNN in order to enhance the performance of sequence search against jitter and stagger. Noon's algorithm has a similar response to the original sequence search algorithm for a stable PRI sequence. However, after removing all stable PRI sequences, and when the remaining sequence is a combination of jitter and noise, the original Sequence-Search is used in order to provide a hypothesis that the given sequence is periodic. This hypothesis is tested by RNN[1].

An isomorphic sequence search algorithm was proposed in [50]. The algorithm utilized a text search tool called a Suffex Array. The Suffex Array finds repetitive parts of a string [50]. The computational complexity of the isomorphic sequence search algorithm is $O(N)$, which is a low complexity.

[1] This algorithm is not clear about the method of establishing a hypothesis from a sequence search.

3.1.2 Time Difference Histogram (TDH)

In this algorithm, time difference (∇TOA) for all pulse pairs is calculated. A histogram is then created from these differences. A histogram is a statistical tool that counts the number of occurrences for a set of observed events within a defined space. The events here are the PRI values, which in turn are the time differences of pulse-pairs (∇TOA). If a pulse-pair consists of pulses from the same radar sequence, then the calculated difference corresponds to the correct PRI or to a multiple of the correct PRI. Conversely, if the time difference was established from pulses of different radar sequences, then incorrect PRI measurement (event) occurs. True PRI events will accumulate with time in their histogram bin to form a peak, whereas the other incorrect PRI events will be distributed across many histogram bins. A histogram can be seen as an autocorrelation of a pulse sequence function. This function can be represented by [9];

$$f(t) = \sum \delta(t - TOA_n) \tag{3.1}$$

where $\delta(t)$ is the unit impulse.

The autocorrelation function of $f(t)$ is [9];

$$h(\tau) = \int_{\infty}^{-\infty} f(t) f(t-\tau)\, dt \tag{3.2}$$

$$= \int_{\infty}^{-\infty} \left[\sum_n \delta(t - TOA_n)\right] \left[\sum_k \delta(t - TOA_k - \tau)\right] dt \tag{3.3}$$

The integral will have non-zero value only when $t - TOA_n = t - TOA_k - \tau = 0$. i.e. when $TOA_n - TOA_k - \tau = 0$

Hence,

$$h(\tau) = \sum_n \sum_k \delta(TOA_n - TOA_k - \tau) \tag{3.4}$$

If histogram bin interval is $[\tau_1, \tau_2]$, then;

$$\int_{\tau_1}^{\tau_2} h(\tau)\, d\tau = \int_{\tau_1}^{\tau_2} \sum_n \sum_k \delta(TOA_n - TOA_k - \tau)\, d\tau \tag{3.5}$$

Harmonics are one of the major problems in the histogram. In order to reduce this problem, non-linear mapping for the histogram output was proposed by [10]. This type of mapping will cause harmonic compression.

The performance of a time-difference histogram degrades in a dense emitter's environment. The computational complexity of a time-difference histogram is in the order of N^2, which is relatively high. This algorithm can be used for offline de-interleaving. It is good for giving an indication about the number of pulse trains that could be successfully de-interleaved. However, a time-difference histogram should be combined with other algorithms in order to complete the de-interleaving process.

3.1.3 Cumulative Difference Histogram (CDIF)

This algorithm combines histogram techniques with sequence search techniques in order to reduce the required computations. Sequence search is more accurate and reliable at identifying sequences than the histogram approach, but at the expense of higher computational requirements. Mardia [11] proposed the CDIF algorithm in order to reduce the required computations in sequence search by the aid of the histogram. The computations in the histogram itself were also reduced by the presented CDIF method.

In the CDIF algorithm, one level of difference is calculated. If no detection occurred, then the next difference level is calculated and added to the histogram, and so forth. If detection happened, however, then the detected PRI value will be searched by means of the sequence search algorithm. If this search was successful, then the detected sequence will be removed from the interleaved sequence, using the sequence search algorithm. Thereafter, the CDIF will reset and start from the first difference level again. In fact, by combining CDIF and sequence search, the performance of both algorithms enhanced.

The CDIF algorithm is intended for stable PRI sequences. Unsorted pulses should be processed using some other techniques. The CDIF algorithm should be implemented after some sort of sorting stage. Sorting is based on one or more single-pulse parameters, such as RF, PW, and AOA.

Detection in CDIF is based on a threshold function. In order for detection to happen in CDIF, both the PRI and its first harmonic should exceed the threshold. Detection in CDIF can be regarded as initial detection. The final detection of a PRI is declared if sequence search was successful for the given PRI.

Jitter CDIF does not extract a jittered PRI sequence reliably. To deal with the jitter problem, Mardia suggested combining histogram bins, such that, the combined bins cover a PRI range that is proportional to the center value of the PRI bin [11]. The sequence search should also allow for more tolerance (in the PRI search) in order to detect jitter. In this algorithm, jitter is only examined after removing stable and staggered sequences from the interleaved sequence.

Staggered CDIF does not de-interleave a staggered PRI sequence, but it can detect the presence of stagger. A staggered PRI sequence will be detected as multiple stable PRI sequences, all will have the same PRI value. The detection of multiple stable sequences with equal PRIs indicates the existence of a staggered sequence. A staggered level value is the same as the number of sequences of the identical PRI. The multiple stable PRI sequences resulting from a staggered sequence then need to be combined somehow in order to allow this algorithm to de-interleave a staggered PRI sequence.

Frequency Agility Mardia suggested using multi-parameter sorting prior to PRI sorting. If the radar sequence has frequency agility, then the sequence will be fragmented into multiple groups, which could lead to the failure of the PRI sorting algorithm. CDIF should function properly under frequency agility condition, however, if frequency is not used as a presorting parameter. Otherwise, the algorithm will be sensitive to frequency agility.

Missing Pulses A high rate of missing pulses leads to either miss-detection or a high false detection rate.

3.1.4 Sequential Difference Histogram (SDIF)

This algorithm is an enhancement of the CDIF de-interleaving algorithm. The proposed algorithm (by Milojević) [12] combines time-based methods with multi-parameter presorting, which is also, proposed by CDIF. SDIF enhances the performance of CDIF with regard to speed, required computations, and reliability. There are two key points in this algorithm: discarding the accumulation of the levels of the difference-histogram, and deriving the optimal threshold. The detection criterion is the reason behind the accumulation of the difference-histogram levels in the CDIF. In order to have an initial detection in CDIF, the PRI and its first harmonic should both exceed a preset threshold. In the SDIF algorithm, the initial detection does not require the presence of the first harmonic and, hence, accumulation is no longer required. Discarding the accumulation of the difference-levels will cause a 50% reduction in the required computations. The use of the optimal threshold will enhance the speed of the initial detection. This is important to prevent making unnecessary calculations that result from forming more difference histogram levels.

The algorithm consists of three main parts: AOA clustering, SDIF, and sequence search. Under conditions of a dense emitter environment, the performance of SDIF degrades. AOA clustering is thus used to reduce the effect of environmental density. Pulses that come from different azimuth angles will be grouped together, based on their azimuth angles. The clustering will be performed

using the histogram technique. We will comment on histogram clustering later. After clustering, the SDIF will be applied to each resultant cluster (group). Thereafter, SDIF makes the initial detection of the PRI values. Each PRI value detected by SDIF will proceed to a sequence search algorithm, which searches for the specified PRI. If the sequence search succeed in its search for a given PRI, then detection is declared (the sequence search algorithm was explained in an earlier section, see Section (3.1.1)). In order to improve the performance of the sequence search, a two-dimensional (2-D) search was proposed [12], where both TOA and PW are involved.

The threshold function proposed by Milojević needs two constant parameters that are determined experimentally. These two parameters are scenario dependent, as Kuang found in his simulation [51]. Threshold parameters are very sensitive to the number of active emitters [51]. Kuang proposed replacing the threshold detection technique by a simple peak search technique. He built his proposal on the assumption that "extreme" histogram values in SDIF correspond to correct PRIs. Moreover, for smaller PRIs, the chance is higher of creating a peak in a low difference-level of SDIF.

Jitter According to Milojević [12], jitter will cause a group of PRI values to appear around the true PRI. Milojević believes that, the amplitude of the PRI bins around the true PRI bin can exceed the detection threshold of SDIF due to jitter. Milojević tried to provide a solution by proposing a check for whether the range of PRI values falls within permitted tolerance. If the PRI values are within the permitted tolerance, then the central PRI value will be searched. Contrary to what Milojević believes, we think jitter will rather reduce the chance that the amplitude of the true PRI and its jitter components will exceed the threshold, because jitter will cause a spreading of the observed PRI value over multiple bins. This spreading will reduce the amplitude of the true PRI in the histogram. The amplitude of the jitter components will be proportional to the reduction of the true PRI amplitude. Jitter components will – generally – be reduced, as spreading of jitter increases. Based on this, we can conclude that the derived optimal threshold will reduce the chance of detecting jittered PRI, and, we thus cannot consider SDIF to be reliable for jitter PRI. Since this algorithm is using the sequence search algorithm, the overall performance (regarding jitter) can be enhanced or degraded, based on the performance of the sequence search. This (sequence search) has been discussed in an earlier section (see section (3.1.1)).

Staggered Milojević obtained successful results with regard to sorting a staggered PRI sequence. However, no details were presented on how this algorithm will be able to handle staggered PRI. The approach provided by the former CDIF algorithm was thus most likely used by Milojević.

Missing Pulses Missing pulses can cause the harmonics of the PRI to exceed threshold, as stated by Milojević [12]. If both, the harmonic and the basic PRI exceed the threshold, then the sequence search will extract the basic PRI value. This is because the sequence search starts from lower PRI values. In this situation, therefore, the effect of harmonic is mitigated. Conversely, if only the harmonic exceeds the threshold, then false detection would occur. Milojević suggested checking for sub-harmonics by finding the "maximum" peak of the histogram. If the maximum peak did not exceed the threshold, then the first lowest PRI exceeding threshold would be checked. If it were found to be the basic PRI of maximum peak, then the sequence search would be performed for this basic PRI value.

Frequency Agility The algorithm utilizes the frequency histogram only for detecting the presence of frequency agility. The frequency thus cannot be considered as a third sorting parameter; conversely to what was introduced. The frequency histogram is applied after both AOA clustering and SDIF. Hence, we can conclude that, the presence of frequency agility will not degrade the performance of the algorithm.

Pulse-Width Agility In order to enhance the performance, a 2-D sequence search is performed, utilizing PW as well as TOA. Therefore, if the interleaved sequence includes PW agility, then the 2-D search will degrade the performance of the algorithm. The algorithm should be able to sort sequence with PW agility if the 2-D sequence search is discarded (at the expense of algorithm performance).

Dense emitters SDIF will not be able to function properly under conditions of a dense emitter environment without using presorting. The presorting in the presented algorithm is based on an AOA histogram. It is clear that the performance of presorting using an AOA histogram will influence the performance of the algorithm under conditions of a dense emitter environment. The performance of the AOA histogram in clustering needs attention. We can conclude that, the algorithm should work fairly well in a dense environment if more attention was given to AOA presorting.

SDIF is usually described as a two-part algorithm that consists of difference histogram and a sequence search as in [52]. Bent considers SDIF to be successful within a dense emitter environment [52], which we agree with if AOA pre-sorting is part of the algorithm.

Noisy measurements The important parameters in SDIF are TOA, AOA, and PW. TOA noise will affect SDIF, as well as sequence search. It needs to be determined how TOA noise

will affect each of the two parts of the algorithm, and what effect PW noise will have on a two-dimensional sequence search. The accuracy of AOA will strongly influence the initial clustering performance. However, there is not enough description with regard to how histogram will be used in AOA clustering.

Corrupted Measurement Milojević has not discussed the effect of a corrupted measurement on his algorithm. It needs to be determined what the effect of multi-path propagation is on each of the three parts of the algorithm. AOA can be affected by multipath. Then, the algorithm could identify the same emitter from different angles. Finally, it is easy to see that a corrupted PW measurement will degrade the performance of the sequence search. Generally speaking, the performance of algorithm against corrupted measurements is good.

Missing points It also needs to be assessed what the effect of resolution of the EW receiver will be with regard to the algorithm, and what the effect of measurement domain of the EW receiver has on the algorithm.

3.1.5 PRI transformation

The PRI-transform algorithm tries to overcome the problem of subharmonics associated with the histogram methods (CDIF and SDIF) discussed in Section (3.1.3) and Section (3.1.4). Subharmonics are the peaks that appear at integer multiples of a true PRI value. Histogram methods rely on the concept of autocorrelation function, which was introduced in earlier section (see Section (3.1.2)). The PRI-transform uses a complex version of the autocorrelation function in order to take into account phase information. Generally, TOA differences with correct PRI value have identical phase components, while TOA differences corresponding to subharmonic values of PRI have different phases. When integration is applied in autocorrelation function using phase information, similar phases tend to add up constructively and to form a peak at true PRI bin, while varying phases tend to cancel out at the PRI subharmonic bins. The following equations demonstrate the difference between the two autocorrelation functions;

$$C(\tau) = \int_{-\infty}^{\infty} g(t)\, g(t+\tau)\, dt \tag{3.6}$$

$$D(\tau) = \int_{-\infty}^{\infty} g(t)\, g(t+\tau)\, e^{(i\cdot 2\pi\cdot t)} dt \tag{3.7}$$

The original PRI-transform works well for constant PRI sequences. However, jitter poses a

serious problem for this algorithm. PRI peaks can be strongly reduced if not completely disappear due to jitter, mainly for two reasons: phase noise growth with time, and spreading of pulse-pair-difference values. Nishiguchi [13] proposed using overlapping PRI bins with wider widths in order to solve the spreading problem. The width of each bin is proportional to the PRI value at that bin and to the maximum jitter ratio. For the phase noise problem, which is a complex one, he proposed shifting the time origin for each bin, using a defined criterion. Phase noise growth is proportional to the number of PRI periods, starting from a given time origin. After a certain number of PRI periods, the time origin should be increased in order to prevent big phase growth. Mahdavi [14] found the optimum time for shifting time origins in order to enhance sub-harmonic suppression further. He also suggested maintaining constant accuracy across all bins by maintaining a fixed ratio r for all bins; where $r = {}^{\tau_k}/_{\tau_{k+1}}$, and τ_k is PRI bin value at bin index k. Furthermore, this bin configuration will reduce required computations.

Guo algorithm [53] addresses extending the PRI-transform range without sacrificing PRI resolution. This algorithm tries to optimize the PRI-transform resolution by limiting the PRI-transform range to the actual range of the received interleaved sequence. This algorithm can only work if the actual interleaved sequence has a relatively limited PRI range.

SDIF is used in Guo algorithm in order to give an indication of the PRI range. However, Guo has not explained clearly how the algorithm will obtain the range from SDIF. It should be noted that, nature of SDIF detection is sequential. Therefore, for every new histogram difference established by SDIF, there is a new resultant range. It is unknown if the range will be calculated from each SDIF difference, or if it will be calculated after certain number of differences, and whether if this number of differences is fixed or dynamic. Moreover, if the number of differences is dynamic, then what is the criterion to determine this number? In the simulation scenario presented by Guo, there were four difference levels in the SDIF. The algorithm can be described as follows: SDIF will determine the range of PRIs, and then the PRI transform will be established based on this PRI range. The constant PRI sequences will be detected by PRI transform and then removed from the interleaved sequence. The sequence search could be used to remove these constant PRI pulses. The number of residual pulses will be monitored. If this number is high, then it gives an indication of staggered sequence existence, and therefore SDIF is applied on the residual pulses. If detection happened on a certain difference level, then this could be an indication of the level of stagger. It is very clear that, this algorithm relies on SDIF in the detection of staggered presence. Stagger detection using this method needs to be evaluated further.

Staggered PRI detection is done by monitoring the number of remaining pulses after de-

interleaving constant PRI sequences. If the number of the residue pulses is high, then a staggered existence can be declared and SDIF will be carried out until detection happens. However it can be argue that Guo algorithm has not provided a solution regarding the de-interleaving of staggered PRI.

In order to solve the problem of jitter in PRI-transform, Ren proposed applying FFT on PRI-transform spectra [15]. FFT is applied only after removing fixed PRI values from PRI-transform spectra. The highest peak of FFT corresponds to the correct Pulse Repetition Frequency (PRF) (the reciprocal of PRI).

Wu presented a smoothing method in order to enhance finding PRI peaks in PRI-transform in presence of jitter [16]. However, the feasibility of using smoothing is questionable, especially when PRI values are closely located in PRI spectra.

3.1.6 Time-Period PRI Transform (TP-PRI-Transform)

Time-Period PRI Transform (PRI-Map) is an extension of the PRI-transform algorithm in order to allow for detection of short time pulse trains, even with PRI jitter. The limitation of the PRI-transform in the detection of short time pulse trains is similar, to some extent, to the case of Fourier Transform (FT) with regard to the analysis of short-time signals. Time-frequency analysis using wavelets is used to overcome this problem in FT. Time-period analysis introduced by [17], imitates the wavelet concept for the PRI-transform. The transform is defined by the same formula that describes the PRI-transform except for integration intervals. The calculations are done within a moving window. The width of the moving window is dependent on PRI bin values. In the PRI-Map method, the update of the time reference is not required anymore, because phase noise growth is already restricted by the width of the moving window. The interleaved TOA pulse sequence is transformed by the PRI-Map method into the PRI spectrum, which is a function of time. In the PRI-map, the concept of detection and threshold is completely new and more complex than the one used in PRI-Transform.

The PRI-Map provides information regarding short length PRI sequences, which can find useful application. In fact, this local PRI information is not necessary for the de-interleaving, especially when associated with the complexity expense. In addition to providing of the local PRI information, PRI-Map enhances the detection of short time sequences that can be missed in the normal PRI-transform. In theory, this can be achieved indirectly by the normal PRI-transform, when it is applied repeatedly after removing each detected PRI sequences.

3.1.7 Hough Transform

Hough transform was used in to find the PRIs of interleaved pulse sequence as presented in [18]. Hough transform is an image processing technique that was originally used to detect straight lines within an image. The intercepted TOA sequence is transformed into a 2-D bit-map image, where each TOA is represented by an (x,y) point. This can be achieved by folding the time axis in manner similar to a raster scan pattern. The result of this process is an image with different straight lines, where points related to same emitter will fall on a single straight line. The slope of the line is linearly related to the emitter's PRI. The task of Hough transform is to detect these lines within the image.

Suppose straight lines passing through a single point are represented by equation (3.8);

$$y = M\,x + K \tag{3.8}$$

where y, and x are respectively the y and x axis coordinates. M is the slope of the line, and K is the vertical displacement. Obviously, one point in a plane has a very large number of possible straight lines that can pass through it. And because x and y are already known, the equation (3.8) can be rearranged as follows;

$$K = y - M\,x \tag{3.9}$$

Now every point is characterized by number of lines each of which is represented by two parameters M and K. The values of M and K will be calculated for each TOA-point in the bit-map (many values of M and K). In this way, a 2-D histogram of (M,K) is established. The peaks in this histogram correspond to straight lines. However, there is ambiguity in determining the PRI from a detected line, because each detected line represents a group of harmonically related PRIs. The bandwidth of the PRI is limited in this method and, therefore, multiple bit-maps are required to cover higher bandwidth.

Because line detection in Hough transform does not rely on a single TOA-point, this method should work well in the presence of missing pulses. The sensitivity of this method for jitter needs more investigation, but generally speaking, the performance can be controlled by quantization factors of M, and K parameters or even bit-map resolution. The method provided in [18] has not provided a solution for the problem of staggered sequences. It will be interesting to investigate the performance of this algorithm under jitter and stagger conditions.

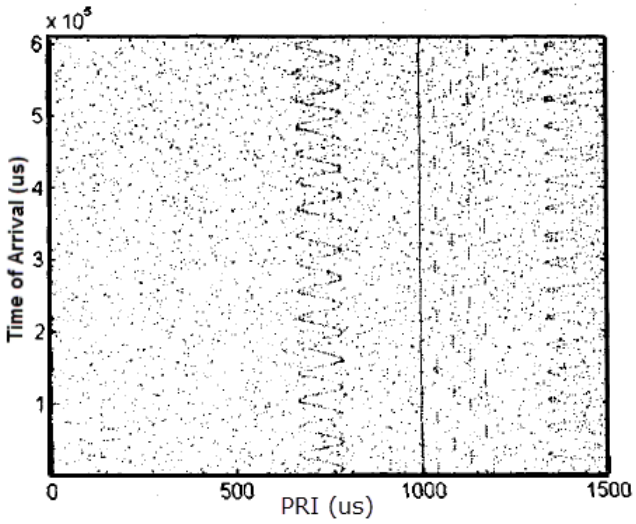


Figure 3.1: Plane transform for three emitters [19].

3.1.8 Plane Transform

This algorithm transforms the TOA sequence into 2-D bit-map, where one axis represents time and the other represents PRI (as shown in Figure (3.1)). For each TOA, all possible PRI values are calculated and added to the bit-map image with respect to time. In order to get rid of noise, the difference matrix is calculated in both forward and backward directions. The true PRI values are correlated in both direction, while noise is uncorrelated [19].

$$\nabla T = \begin{bmatrix} T_{11} & T_{12} & .. & T_{1M} \\ :: & T_{22} & .. & T_{2M} \\ :: & & 0 & : \\ T_{N1} & .. & . & T_{NM} \end{bmatrix} \tag{3.10}$$

The scenario presented by [19] in Figure (3.1) shows how complex PRI modes can be exposed using this method. However, it is noticed that the harmonics of all PRIs chosen in the figure are greater than the maximum value of the PRI axis, except for the sinusoidal PRI. The mirror (harmonic) of sinusoidal PRI is clear in the figure. In fact, if an emitter with a lower PRI value is added to the scenario (50 us for instance), then the harmonics of this emitter will complicate the image. The situation becomes even worse if more of such emitters are added. Jian suggested using a simple mirror-image removing technique to solve this problem, but this does not seem to work well under plenty of harmonics. It can be seen that this algorithm can work well only with a restricted PRI bandwidth.

In another algorithm,Wei [54] provided a simpler plane transformation technique, where the

pulse sequence is transformed into a bit-map. The coordinate (x_n, y_n) of the n-th pulse within the plane is given by:

$$x_n = \text{round}(\text{mod}(L(n), W)) \tag{3.11}$$
$$y_n = \text{floor}(L(n)/W)$$

where, $L(n) = \sum_{i=1}^{n}(t_i - t_{i-1}), t_i$ is the TOA of the pulse of the index i, and W is the width of the plane.

A pulse sequence of a given PRI will generate periodic patterns within the bit-map. Wei proved that, fixed, jittered, staggered and sinusoidally modulated PRI sequences produce such a periodic patterns within the bit-map. Therefore, he proposed searching these patterns in order to detect the PRIs of the de-interleaved sequence.

3.1.9 Cumulative square sine-wave interpolation algorithm

This method estimates the jittered PRI signal [20], by utilizing the cumulative square sine-wave interpolating (CSSI) algorithm. In terms of this approach, each TOA difference value is modulated with a sine wave that has a period equal to the TOA difference. The modulated signals will be added in order to form a time domain signal. FFT is then applied to the signal. The resultant spectrum will show a peak on the estimated PRI value. If PRI is jittered, then the spectrum around the peak will be broadened, and the peak amplitude will be reduced. The use of Fourier transform is in order to allow for extracting the periodic component of the TOA differences. As noted here, the FFT was not applied directly to the TOA difference signal, but it was applied on its sine wave interpolation version instead. A search algorithm will be applied to the sequence in order to extract the detected PRI sequence from the whole sequence. This process will be repeated for the remaining sequence samples. The advantage of this algorithm over the Difference histogram, CDIF, SDIF, PRI-transform, and plane transform is its ability to sort jitter PRI accurately compared to those methods. This algorithm can also sort fixed and jittered PRI sequences.

The simulation provided by [20] showed that an almost 90% correct sorting rate was achieved. This was done using three simulated overlapping sequence, two of which belonged to jittered PRI, while the other was fixed. However, the behavior of this algorithm at presence of staggered PRI sequence needs to be examined. There are other points that need to be investigated for this algorithm too, such as its performance in dense environment, and the impact of noisy measurements. The implementation of this algorithm also needs to be investigated.

3.1.10 Active Cell TOA De-interleaving

Mardia [55] proposed an identification process that consists of two tasks, grouping (i.e. clustering) and TOA de-interleaving [55]. In the clustering stage, the active cells will be identified. Cells within a specific distance will form a cluster. The parameters used in distance calculation are AOA, RF, and PW. The highest weight is given to AOA, and the lowest is given to PW in distance calculation. Two distance thresholds are used, i.e. one is for stable radars, while the other is for agile radars. The stable threshold is used first for clustering. Then, TOA de-interleaving is applied on resultant clusters. After TOA de-interleaving, the agile distance threshold will be applied on unsorted pulses in order to cluster agile radars. TOA de-interleaving will be applied again and so forth. In the clustering stage, cells are sorted according to their density. Then the Leader algorithm is used to enhance computational efficiency. The Leader algorithm assigns the first cell as leader, and the distance from leader to the subsequent cell is calculated. If the distance is within a determined threshold, the cell will be added to the cluster. Otherwise, the cell is identified as a new leader. The distance between new leader and the subsequent cell will be calculated and so forth. Clusters with fixed distance will be formed. Clusters within a specific distance from each other will be combined, if the density of clusters is lower than a certain threshold.

3.1.11 Time Progressive Processing

In this algorithm [56] [57] three different snapshot lengths of PDWs are processed. The length of the first snapshot is 5 ms, the second snapshot is 30 ms, and the third snapshot is 80 ms. Other lengths are possible, such that, each snapshot length is smaller than the next. The PDWs in the first snapshot are processed by an algorithm. The remaining unsorted PDWs are removed and placed on the second snapshot. Another algorithm is applied to the second snapshot. Then, again, the unsorted PDWs are removed and placed in the third snapshot, and, another algorithm is applied on the third snapshot.

The first algorithm is intended to check if any previously detected signals are present in the 5 ms snapshot. The second algorithm is used to detect PRI in the 30 ms snapshot. The algorithm used is the Differential Time of Arrival Histogram (DTOA); (also designated as TDH in literature review). Hough transform is also used in order to detect PRI patterns that cannot be detected by means of DTOA. The third algorithm is also intended to detect the PRI in the 80 ms snapshot, and this algorithm is based on a spectrum estimation technique.

A clustering stage is used before PRI detection in the second and third snapshots. The clustering

algorithm used is Iterative Self Organizing Data Analysis (ISO-DATA). This is an unsupervised pattern recognition algorithm,which is used for classification. As the name indicates, this clustering algorithm is iterative. In each iteration, the pulses in the snapshot are clustered. The result of clustering is examined using a similarity test in order to determine if there is a need to split or to remove some of the clusters. This iterative process continues until no splitting or removal of clusters is possible, or until the number of maximum iterations is reached.

The similarity test used in the clustering algorithm is based on Euclidean distance. The degree of similarity is measured by a performance index, such as the sum of the squared error of the cluster.

3.1.12 Multiple Correlation Hypothesis

This algorithm [58] [59] is used to target stable PRI emitters. De-interleaving stable PRI sequences will reduce the burden on the limited processing resource in order to allow them to concentrate on more complex PRI patterns.

The algorithm employs linear detection and linear regression in parallel. Then, a dual vote is used in order to confirm the estimated PRI.

In linear detection, the TOA vector is normalized. The coarse PRI is estimated. Then, a number of PRI trial values around the coarse PRI are assigned. Based on a trial PRI, a correlation vector is constructed and compared with the normalized TOA vector. Subtraction of the two vectors is performed in order to obtain a weight vector. The correlation result is the sum of this vector, normalized to span of the PRI trials. Then the PRI of best correlation value is selected.

In linear regression, an error function is calculated, utilizing three other obtained quantities viz., Mean PRI estimate, mean estimate of TOA, and a weight vector. The error function should be within a predetermined threshold in order to declare a detection.

The dual vote simply combines the results of the linear detection and the linear regression in order to provide the final detection results.

3.1.13 Auto-correlation

There is a similarity between this algorithm and CDIF and SDIF. The first step in this algorithm [60] is involving sorting received pulses into groups based on RF and AOA windows. Each group therefore has pulses of a similar angle and frequency. After that, for each group, auto-correlation is applied on the pulse sequence in order to detect PRIs. For each detected PRI, an "integrator" is established. This integrator is a form of histogram that shows all TOAs relative to a detected PRI. The pulses of the emitter with the correct PRI will accumulate in the same integrator cell, creating a

peak. Thereafter, detection is done by a threshold. In addition, peak detection and other detection techniques are utilized to detect complex peaks, which result from jitter and multiple emitters.

3.1.14 Pandu Algorithm

Pandu [61] proposed a de-interleaving algorithm that uses combination of a TOA sorting method and a parameter clustering method in order to enhance the de-interleaving performance. The main idea of this algorithm to use a histogram for clustering, based on the AOA parameter. Hence, AOA is the first sorting parameter. TOA sorting is applied thereafter for each resultant AOA cluster. The TOA method used was not indicated, but it is most likely the CDIF method. A frequency histogram will be applied after that on the sorted PRI pulses. The frequency histogram is used only to detect frequency agility within the sorted sequence. Finally, sequence search will be applied to the sorted PRI sequence. Pandu suggested using 2-D search, where PW is utilized along with PRI. However, no details are provided about how this 2-D search will be approached. There is no indication of the performance of this algorithm, which is almost identical to Milojević's algorithm [12]. Pandu proposed an FPGA implementation for this algorithm.

3.1.15 Folding Algorithm

This algorithm [62] can be described as follows: Firstly, it selects a random "mid" point from the pulse sequence. The pulse sequence can be represented by impulses that are located on the time axis, at TOA values. The time axis is then folded around the "mid" point. The impulses that are related to the same emitter will be superposed after the folding of the axis. The algorithm, according to [62], is fast, accurate, and robust to noise. However, the performance of the algorithm for jittered PRI sequences is unknown.

3.1.16 Other algorithms

In [63], Per proposed a method in order to distinguish between real and false emitters that appear at receiver input. One stage of this method is de-interleaving. However, the only information provided about this de-interleaving stage is that, it is based on TOA, and on sorting based on AOA.

In 1988, Jenkin [64] proposed a real-time pulse processor, which used a folded shift register hardware in de-interleaving. The folded shift register was utilized to detect pulses at a specific PRI. In fact, it is an old system, which is mentioned here for the sake of completeness.

After reviewing the various time-based de-interleaving algorithms that are available in the literature, the next section reviews the multi-parameters clustering algorithms used in de-interleaving.

3.2 Multi-Parameter Clustering

These algorithms rely primarily on the single-pulse parameters, excluding the time parameter. This review covers various algorithms, which include: Self Organizing Feature Map (SOFM), Support Vector Clustering (SVC), Kernel Fuzzy C-Means (KFCM), Fuzzy-ART (F-ART), K-means, Anchor Clustering, in addition to various other algorithms.

3.2.1 Self-Organizing Feature Map (SOFM)

SOFM is a method that belongs to Artificial Neural Networks (ANN). Four algorithms were identified in the literature as using SOFM for de-interleaving radar pulses. These four algorithms address three different problems, viz., firstly, is the prior knowledge of the number of the radars, which is required by SOFM; secondly, the change of training patterns of SOFM due to radar parameter agility; thirdly, enhancement of de-interleaving sorting rate using SOFM.

The first algorithm uses a validity index called Composed-Density-Between-and-Within-Clusters (CDBW) in order to find the right number of clusters to be used within SOFM, and with lower computational complexity [21]. The second algorithm modifies the distance measurement method in SOFM in order to suit the radar environment, and in order to prevent degradation of the SOFM clustering ability due to the agility of some radar features [22]. The last two algorithms use SOFM for clustering based on radar parameters that are extracted from wavelet features, and from resemblance coefficients [23] [24].

SOFM-CDBW Some clustering algorithms require the number of clusters or even the center of clusters in advance. In fact, such information about clusters is usually unknown in the case of ELINT. Algorithms that do not require prior knowledge of clusters usually involve high computational complexity, which makes them impractical for large data [21]. Dai [21] used a validity index called Composed Density-between-and-within-clusters (CDBW) with SOFM in order to find the right number of clusters, and did so with lower computational complexity. There are many validity indexes, which rely either on measuring the compactness of clusters or on measuring the separation of clusters. The CDBW index was used by [21] because it relies on both measures.

Initially, SOFM is established from a number of nodes that is bigger than the expected number of clusters. This will lead to dead nodes, which are not used. Using CDBW, these dead nodes are

eliminated. In order to find the optimum number of clusters, SOFM will be applied several times, each time to a give different number of clusters. The number of optimum clusters is the number that corresponds to the highest CDBW value. The number of clusters begins with a high value, which is then reduced by merging of the clusters. The clusters with the lowest CDBW index, and with direct neighbor relationships are merged with each other. The number of clusters will continue to be reduced in this way, until only two clusters remain or until no more clusters can be merged. SOFM-CDBW was applied for azimuth and elevation AOA parameters.

The elevation AOA parameter is not measured in most EW receivers. Furthermore, replacing the elevation AOA by another parameter for this algorithm will not necessarily give the same performance. The algorithm was tested for nine emitters with relaxed separation between clusters, which makes sense for azimuth-elevation clustering. However, this clustering method may be unsuitable for other pairs of parameters.

Modified Distance SOFM Some radars have agility in their parameters. The change of radar parameters in a training pattern can prevent SOFM from performing correct clustering. The use of Euclidean distance in SOFM can show a long distance between two samples related to same class, as in the case of frequency agile emitters.

The distance measurement definition is modified by [22] as in equation (3.12) in order to suit the emitter's environment. This distance calculation may not even be necessary if the AOA separation is larger than a specified threshold. The AOA parameter is used because it is the only parameter that is independent of the radar signal. Discarding unnecessary distance calculations will enhance the convergence speed.

$$d^{'}\left(x^{k}, m^{j}\right) = \begin{cases} \sum_{n=1}^{N} \alpha_n \left| \frac{x_n^k}{max\{x_n\}} - m_n^j \right| & max\{x_n\} \neq 0 \\ 0 & max\{x_n\} = 0 \end{cases} \tag{3.12}$$

where; n: the index of the parameter within a pattern, k: the index of the pattern, m^j: the weight of the output layer, x_n^k: the parameter with index n in the pattern x^n, and m_n^j: the weight between the parameter of the index n and the jth cell of the output layer. The value of the vector α_n is experimentally obtained, which means that, this constant is scenario dependent. However, in order to make this distance a useful one, the constant vector α_n should have some extent of generalization. In fact, only four radars were used by [22] in order to test this algorithm.

Wpt6-Cr1- SOFM In order to enhance the sorting rate and reduce the sensitivity to SNR, the resemblance coefficient and the wavelet package characteristic of the radar pulse are used in

radar clustering [23]. The wavelet package characteristic is eight dimensions, and the resemblance coefficient is two-dimensional. Only two parameters are used, namely, Cr1 from the resemblance coefficients, and Wpt6 from the wavelet package characteristic. Using more parameters will increase the computational complexity. The two parameters were chosen by [23] because they are good in the calculation and in terms of their degree of separation. Different types of radar signal modulation can be classified effectively by means of the resemblance coefficient, even with low SNR. Moreover, SOFM is used to cluster the radar pulses based on these two parameters.

Wpt6-Wpt7- SOFM AOA and PW are used with Wpt6 and Wpt7 in order to enhance the sorting rate [24].

The nature of the ELINT input data structure is dynamic. SOFM has two stages: firstly, the training stage, where maps are established based on input data structure, and secondly, the mapping stage, where automatic clustering is performed [65]. In fact, SOFM is suitable for a static input structure and will not be able to learn new data structure once the training stage is finalized [66]. Hence, SOFM is not suitable for on-line ELINT application. In fact, there is relatively little researches on on-line SOFM. However, [67] proposed the use of a Dynamic Self-Organizing Map (DSOM) for cortical plasticity application, which could also find an application at the de-interleaving.

3.2.2 Support Vector Clustering (SVC)

Four types of algorithms were identified using Support Vector Clustering to sort radar pulses. All of the proposed algorithms addressed the computational complexity of SVC. The algorithms are Cone Mapping Support Vector Clustering (CMSVC), Delaminating Coupling based on SVC (DCSVC), SVC and K-Means with Type-Entropy, and On-line Independent Support Vector Machines (OISVM).

CMSVC In this algorithm, the Cone Cluster Labeling (CCL) method is used with SVC [25] in order to reduce time complexity of clustering. The clustering is based on RF, PW, and AOA. A validity index was also introduced in [25] in order to make it possible to evaluate the compactness and separation of the clusters. This validity index is based on the information entropy theory.

DCSVC The number of samples for radar pulse sorting is very large compared to the data size that SVC essentially deals with [26]. Training using all samples will produce a very large adjacency matrix for SVC, which thus reduces the speed of the calculations. Li and Guo [26]

[27] [28] [29] proposed splitting pulse trains into short sequences. Each short sequence is then clustered using SVC and centroids of the clusters are found. Thereafter, the resultant centroids of the different short sequences are clustered again by mean of SVC. Single radar may exist in several short sequences, and therefore clustering of centroids links related clusters of different short sequences to each other. The complexity of SVC is $O\left(N^2\right)$ [26], and the splitting of the pulse sequence into short sequences will enhance the speed of the computations. In [28], type-entropy recognition is used in conjunction with DCSVC to assist the sorting process further. In information theory, source entropy reflects the uncertainty or the amount of information provided by a message from an information source. Accordingly, type entropy is used by [28] in order to describe the complexity of the signal environment, based on the characteristics of the radar environment. This information is used in setting some DCSVC parameters that control the boundaries of clusters [28]. The above algorithms were proposed in order to comply with the real-time need of radar pulse de-interleaving. It was shown by [30] how DCSVC is able to overcome the problem of tolerance setting of conventional sorting algorithms. The parameters that were used in the clustering are RF, PW, and AOA.

K-means-SVC The K-means is used with SVC by [68] in order to reduce the complexity of SVC in de-interleaving radar pulses. Type entropy is used to reflect the complexity of the environment. The higher the value for type entropy, the more complex the environment, where the highest value is $N \cdot log\left(N\right)$ and N are the categories of pulses. SVC is used as an initial stage and applied to a small pulse sequence sample, and type-entropy is used to tune the SVC parameters. The initial SVC predicts the centroids of clusters that will then be used by the K-means algorithm. In the K-means algorithm the initial centroids and the number of clusters are provided in advance. If this information is not correct, then false alarms are generated. SVC in this case will predict this prior information in order to provide it to the K-means clustering algorithm. SVC searches for the global optimum solution and has good generalization, but the complexity of the SVC algorithm is high $O\left(N^2\right)$ [68]. Therefore, SVC cannot be used for large data. Conversely, the complexity of the K-means is $O\left(k \cdot N\right)$. Hence, combining SVC with K-means can make it possible to conduct real-time sorting of radar pulses [68]. The parameters used in clustering in this algorithm are; RF, PW, and AOA.

OISVM SVM uses batches of data in its training process. The size of the SVM solution increases linearly with the number of samples [69]. Moreover, SVM requires retraining whenever new sample is received. Therefore, standard SVM is impractical for application in radar pulse de-

interleaving. Zhu [69] shows that using OISVM as a classifier for intra-pulse features can achieve good accuracy and on-line learning too. The intra-pulse features used are: binary phase shift keying (BPSK), nonlinear frequency modulation (NLFM), binary frequency shift keying (BFSK), quadri-phase shift keying (QPSK), and conventional pulse (CP). The type of the modulation is found based on an instantaneous frequency analysis of the pulse.

3.2.3 Kernel Fuzzy C-Means (KFCM)

The resemblance coefficients Cr2 and the wavelet dimension Wpt4 were used with Kernel Fuzzy C-Mean (KFCM) by [70] in order to increase the correct sorting rate. These two parameters were selected because of their good separation degree in classification and their low sensitivity to noise. Resemblance coefficients are good in identifying the intra-pulse modulation mode, which is dependent on the spectrum shape of the pulse. KFCM is also used by [71], but using wavelet dimensions Wpt2 and Wpt5.

Wang [72] used KFCM with resemblance coefficients. These coefficients are based on bi-spectrum 2-D characteristics. These characteristics have good separability and and good immunity to the noise [72].

Yu [73] used Higher-order statistics (HOS) parameters with KFCM; these parameters are used in order to classify signals into different modulation types. However, HOS cannot sort signals with the same modulation, and hence the PRI-transform is used after KFCM [73].

3.2.4 Fuzzy-ART (F-ART)

Fuzzy-ART has some advantages like lower computational complexity and small memory requirement [31]. Ata'a [32] proposed using Fuzzy Adaptive Resonance Theory (Fuzzy-ART) in de-interleaving. This algorithm can sort a high number of emitters simultaneously and with a high correct sorting rate [32]. In addition, this algorithm is suitable for real-time application due to its high learning rate.

Noone [31] proposed a modification to Fuzzy-ART in order to allow it to use prior information regarding measurement accuracy. However, using prior information in Noone's algorithm does not mean that the algorithm is supervised.

Thompson [33] presented a test for a Fuzzy-ART implementation on an application specific integrated circuit (ASIC). Fuzzy-ART was compared with K-means clustering, Fuzzy-Min-Max clustering (FMMC), and Integrated Adaptive Fuzzy clustering (IAFC) in [33]. Fuzzy-ART was found to be the fastest and with the highest correct sorting rate using the available data set.

Cantin [34] proposed VLSI implementation for the fast clustering of radar pulses.

3.2.5 K-means

Zhifu [35] performed AOA sorting as an initial stage. Thereafter, K-means clustering was performed, based on the RF and PW for each AOA cluster. Finally, a sequence search was used to extract the sequences. In the original K-means algorithm, K random objects are assigned as initial centroids for clusters. Zhifu suggested improving the method of initializing the centroids by selecting pulses that have different parameter values. In fact, this does not seem to achieve significant improvement to centroids initialization process. Furthermore, the number of clusters (K) is unknown, which makes it a serious problem for this algorithm.

3.2.6 Anchor Clustering

In this algorithm, the anchor point concept is utilized in clustering [74]. The anchor point defines a cluster. In anchor point clustering, the first received pulse becomes an anchor point which defines the center of a cluster. The similarity between the anchor point and other pulses is measured based on weighted distance. If a subsequent pulse is sufficiently similar to the anchor point, then it will join the cluster. Alternatively, it will be assigned as a new anchor point (new cluster).

In distance calculation, the weight of each pulse parameter is inversely dependent on measurement error. These weights are calibrated regularly by the de-interleaver. This is achieved by injecting pulses into the input of the receiver and adjusting the weights based on receiver output.

3.2.7 Other Methods

Blind Source Separation (BSS) Blind Source Separation (BSS) is originally used in audio processing to separate an audio source from a set of mixed sources. BSS is a method for recovering a source signal from a mixture of statistically independent signals that are received from different sources. In BSS, there is no prior knowledge about sources or about the mixing process [75].

BSS was used to de-interleave radar pulses, as shown by Keshavarzi [75]. Keshavarzi assumed the number of sources had already been estimated, using a method based on dynamic decorrelation. The sources and sensors are assumed to be fixed in this algorithm.

Independent component analysis (ICA) was used in de-interleaving by Amishima as indicated in [36]. The problem with this technique is that, it requires the number of the sensors to be at least equal to the number of the sources. Sparse component analysis (SCA) is another BSS technique

that is used by [36] in order to deal with the situation where the number of sources is more than the number of sensors. He used the sparsity of intra-pulse modulation of radar signals.

Spectrum Atom Decomposition and Kernel Method Parameters of radar are extracted based on the time-frequency atom [37]. The signal is decomposed using Matching Pursuit (MP) based on FFT. Then atom characteristics vector, which has good clustering properties, is extracted. Kernel clustering is used thereafter. This algorithm can separate emitters with overlapping parameters in AOA, RF, and PW with relatively low SNR.

Minimum Description Length (MDL) Liu [38] proposed a type of model-based clustering that uses Minimum Description Length (MDL) criterion. Clusters are updated when new data arrives, by merging or splitting the clusters. This algorithm is used for on-line application and is based on former off-line MDL, presented by [76].

This algorithm was compared with an on-line competitive learning neural network algorithm based on a winner-take-all criterion, and found to be outperforming the competitive learning algorithm [38].

State Machine Library This algorithm uses a library of state machine that is based on a library of known pulse patterns [77]. This library is in turn derived from a library of known radar emitters. The algorithm uses one or multiple PDWs to determine the plurality of the state machines that may match the received radar patterns.

This algorithm de-interleaves and classifies radar pulses. It was designated as Concurrent Classification and De-interleaving Machine (CCDM) [77]. The details of this algorithm are not provided, except at a high level of abstraction.

Pulses Parallel Scoring This algorithm [78] uses scoring of pulses. The scoring is based on the distance between pulses regarding a specific parameter. The oldest pulse is compared with all new pulses in order to assign scores. After that, the next oldest pulse is compared against all new pulses. This process is done in parallel for the different pulse parameters.

AOA Sorting This simple de-interleaving method is proposed as part of a data collection system [79] which is carried on a platform like a satellite that has limited computational resources. The system thus sends information to remote sites for further processing. The proposed de-interleaving algorithm for this system is AOA sorting. This de-interleaving can be done using the limited resources on the platform. In fact, the proposed AOA sorting is regarded as the first stage of the

de-interleaving. There are no details provided, however, on how the AOA clustering is approached.

Adaptive Probabilistic Neural Network (APNN) The APNN algorithm [80] generates a probability density function estimation. Sorting is based on the probability density function that is generated for each operating cluster.

Associative Hierarchic De-interleaver In this algorithm [81] [82], similar pulses are clustered into groups. Then hypothetical pulse train models are obtained from these groups. The confirmed hypotheticals are then used to de-interleave their relevant pulse sequences.

Radial Basis Function Neural Networks Yongqiang [39] used a Log-Sigmoid function as the bases function for neural networks instead of a Gauss function. The use of a Log-Sigmoid will enhance stability and learning ability of neural network.

Immune Artificial Neural Network In order to overcome problem with the convergence speed of other ANNs, such as ART and SOFM, Ting [40] used an ANN that is based on an immune algorithm used in biology.

Big Sample Statistic This algorithm uses the biggest possible data sample in the de-interleaving [83]. It relies on Rough Set Theory (RST), which has applications in data mining.

3.2.8 Older Methods

Parker Algorithm In 1992, Parker [84] proposed an algorithm based on the technology available at that time. The proposed solution contains a PDW filter bank in order to block friendly and low priority emitters. Thereafter, clustering based on AOA and RF histogram was performed. The de-interleaving was done in two steps: the first was for stable RF, and the second was for agile RF [84]. It is not clear how the algorithm will identify the agility of the RF for an emitter, however. The AOA used consists of the azimuth and elevation parameters, which could make it easier to identify the RF agility of an emitter. PDWs are taken from an IFM receiver, which is not appropriate for current systems. Content Addressable Memory (CAM) is presented as a possible solution for the AOA and RF histogram [84]. CAM simultaneously compares the input with all cells of memory. The outputs of CAM are the addresses of matching cells. Parker provided only a high-level description, however, and not enough details are available. CAM is also used later on by Driggs [85]. The received signal in Driggs algorithm is divided into time slices. For a given

time slice of received signal, one of the signal parameters is determined. The value of determined parameter is then introduced to CAM.

Space-Time This algorithm [86] utilizes multiple receivers. Each signal reception event is represented by a space-time coordinate. The space component is based on the location of the receiver, while the time component is TOA.

In this algorithm, a subset of four signal events is selected. Thereafter, the time-space Euclidean distances are calculated for a combination of selected events. Then, based on these distances, the Cayley-Menger determinate is calculated. If the result of the determinate satisfies a predefined condition, then the four events are related to the same emitter. Otherwise, another four signal events are selected in order to repeat the previous steps.

Roe Algorithm Roe [87] proposed knowledge-based de-interleaving in 1990. He suggested using prior information such as platforms that exist, their locations, intentions, etc.

Whittall In 1985, Whittall [88] highlighted some challenges associated with de-interleaving, and suggested using a filter to block the PDWs of sorted emitters in order to reduce the processing load on the de-interleaver.

Anderson In 1988, Anderson [89] proposed using ANN in the clustering of emitters, in the case of large data inputs and prior knowledge of emitters.

Chandra Chandra [90], in 1988, highlighted the problem of changing AOA due to the fast movement of emitter platforms. In 1992, he proposed an algorithm [91] to sort radar pulses based on AOA and RF. The clustering is done by comparing the distance to the mean value with a predetermined tolerance (in other words, a threshold). The pulses with distances below the threshold are clustered within the same group, otherwise a new group is established. The mean values and the tolerances for each cluster are updated when a new sample joins the cluster. However, very few details are available on this algorithm. The initialization process of clusters was not mentioned in this algorithm.

Baumann In 1985, Baumann [92], proposed an algorithm that applies Walsh transform on pulse amplitude samples in order to obtain spectral signatures. These spectral signatures are then correlated with the stored signatures of previous detections.

3.3 Combined Algorithms

Chan [41] proposed clustering pulses into cells in a three-dimensional space that consists of AOA, RF, and PW. A new pulse will join a cell if it falls within some defined Euclidean distance from that cell. Otherwise, a new cell is established. The distance is set such that, the separation in AOA, RF, and PW should not exceed the double of the parameter accuracy for each parameter. The next step is to find the confidence level of each cell, using the TOA parameter of cell's pulses. This is done by using a simple time-difference histogram. The histogram is then normalized based on its highest peak. All peaks that do not exceed 0.8 in the normalized histogram will be dropped. Chan supposes that observation time should equal the number of events, multiplied by the corresponding PRI. Observation time of original histogram is compared with the new histogram in order to provide the confidence level.

However, the method used in order to establish the cells is not reliable. This is because cells can grow to form very big cells that contain pulses from different radars. Furthermore, the order of the pulses in the sequence will influence the size and shape of the cells. In addition, the algorithm has not provided a solution in order to link cells that belong to the same radar (radars of agility in RF or in PW). The use of TOA parameter in this algorithm is only done in order to provide a sort of confidence level for each cell.

3.4 Review

Time-based algorithms can de-interleave frequency agile radars, but it has problems when de-interleaving jittered and staggered radar sequences. Conversely, clustering algorithms can de-interleave a sequence that contains jitter or stagger, but not a sequence that has agility in either frequency or pulse-width.

Some proposed clustering algorithms rely on some of intra-pulse modulation parameters such as wavelet package and resemblance coefficients instead of on common PDW parameters (as in [23], [70], [24], [70], and [71]). In addition, HOS was used to derive intra-pulse parameters, as in [73] and [72]. However, these kinds of algorithms do not provide a solution for the agility problem. Moreover, calculating these intra-pulse modulation parameters will add to the processing burden on the receiver. In order to calculate these intra-pulse modulation parameters, the IF samples of pulses should be processed. In such a case, the IF should be processed either in the DSP unit of the EW receiver, or the IF signal should be fed into a new digital processing module. However, it is desired to use only the common PDW parameters in the de-interleaver in order to avoid any

integration complexity with available EW systems.

There are algorithms that combine clustering with time-based algorithms such as algorithms presented by Mardia [55] and by Yu [73]. Mardia's algorithm performs clustering based on Euclidean distance, using AOA, RF, and PW parameters. Then a time-based method (not specified) is applied to the resultant clusters in order to de-interleave non-agile emitters. It was not mentioned by Mardia, however, how to distinguish between stable and agile emitters (in order to de-interleave non-agile emitters). After non-agile emitters have been extracted, clustering is performed again for the remaining pulses in order to cluster agile emitters. A higher distance threshold is applied. It is hard to evaluate this algorithm without knowing the logic used to extract non-agile emitters during the time-based de-interleaving phase. In Yu algorithm, the emitters are clustered based on intra-pulse modulation parameters. This algorithm was evaluated against some digital modulation types. The performance for other types of modulations such as chirps is unknown. However, Yu algorithm cannot sort emitters that have the same type of modulation. Therefore, Yu used PRI-transform in order to de-interleave emitters that have the same modulation type.

The performance of time-based algorithms degrade strongly in dense environment. Therefore, almost all time-based algorithms use pre-sorting stage in order to reduce pulse density. Pre-sorting is performed based on one or more of the parameters; AOA, RF, and PW.

3.5 Summary

De-interleaving algorithms can be classified into three groups, i.e. time-based algorithms, clustering algorithms, and combined algorithms. The most important time-based algorithms include sequence search, which randomly searches for correct PRI and then extracts the sequence. Histogram methods detect the PRIs of the interleaved sequence by counting the number of time-difference events for each PRI. However, histogram methods suffer from sub-harmonics of PRIs, which create ambiguity problems. This is exacerbated in dense emitter environments. CDIF and SDIF calculate histograms cumulatively and sequentially in order to reduce the problem of sub-harmonics to some extent. In order to enhance sub-harmonic suppression even more, PRI-transform extends the histogram method by making use of the phase information of pulse sequences. Hough transform detects straight lines of the PRI bit-map image, where the PRI value can be found from the slope of the line. The Plane transformation algorithm builds a 2-D image from TOA differences. This method is useful for visual analysis, where the modulation of the PRI can be visually interpreted. The time-period analysis algorithm mimics time-frequency analysis in order to provide the PRI spectrum as a function of time. Finally, the CSSI algorithm modulates each TOA difference with

a sine wave that has period and amplitude equal to the TOA difference. Modulated signals are combined in the time-domain before applying FFT. This algorithm does not seem to work with sequences of multiple PRI values.

These time-based methods do not work well in dense environments. Stagger and jitter are challenges for these methods and for PRI-transform in particular. The time-period algorithm performs better than the PRI-transform with regard to jitter, but at the cost of greater complexity. Moreover, in theory, the performance of the PRI-transform for fixed PRI is better than the time-period algorithm. All time-based methods use sequence search in order to extract detected sequences.

The most important clustering algorithms include SOFM, SVC, KFCM, and F-ART. The problems with these methods in general include the need for prior knowledge about the environment, even in the case of unsupervised algorithms, such as SOFM. In addition, the complexity of the algorithms is also problematic; this was addressed in some algorithms, such as CMSVC, DCSVC, and KFCM. Some of clustering algorithms used intrapulse features instead of common PDW parameters in order to enhance the clustering results. The HOS algorithm is an example of these algorithms (it is one of the KFCM algorithms), but this algorithm is unable to distinguish between emitters of the same modulation type.

Finally, the combined algorithms use the TOA parameter in conjunction with other parameters in the de-interleaving process. There are few algorithms classified as combined algorithms in this dissertation. They include Mardia's algorithm [55] and Chan's algorithm [41]. Both used Euclidean distance in clustering, and clustering plays a major role in the de-interleaving process. Time based methods that use a pre-sorting stage (some sort of clustering) could be considered as combined algorithms. However, time-based algorithms are primarily based on TOA.

In this chapter, the literature was reviewed with regard to the algorithms that are related to the de-interleaving of radar pulses, and in particular to on-line ELINT application. In Chapter 4, the proposed de-interleaving algorithm is introduced.

Chapter 4

Proposed Algorithm

The proposed de-interleaving algorithm is presented in this chapter. Firstly, the chapter illustrated the need to combine the use of time parameters (time-based methods) along with some other single-pulse parameters (clustering). Secondly, the alternatives of using which stage (i.e. time stage or clustering stage) before the other are discussed. This discussion gives some insight into the feasibility of using both of a time stage and a clustering stage. The algorithms (from the reviewed literature) that can be used in either the time stage or the clustering stage were discussed.

The System Overview section (Section (4.4)) presents the proposed de-interleaving system. This system is composed of different building blocks, which include, Clustering, Pulse Sequence Segmentation, Agility Resolving, Cluster's Continuity Detection, PRI Analysis, PDW filter, and SNR Filter. None of the clustering algorithms (available in the de-interleaving literature) was found to be satisfactory for our clustering requirements. DBSCAN is a clustering algorithm that has not been used before in the de-interleaving of radar pulses (to the best of our knowledge), although it satisfies all of our desired clustering features. Therefore, DBSCAN algorithm is used in the clustering stage of the system. The general concept of DBSCAN is illustrated in Section (4.4.1.1). The flow diagram of the applied clustering algorithm is presented. Pulse Sequence Segmentation is plays an important role in the management of the input buffer. This block contributes to the enhancement of the algorithm's performance (speed), as described in Section (4.4.2). After that, the Agility Resolving Block is presented in Section (4.4.3). This block consists of two sections, viz. Agility Hypothesis Generation and Agility Hypothesis Test. The time parameter is utilized to resolve agility. Some related statistical analysis is provided within this section (Section (4.4.3)). The Tracking Detection block is presented in Section (4.4.4); this block contributes to the performance (speed) of the algorithm.

4.1 Introduction

As discussed in a previous chapter, sequences that have time agility are problematic for time-based algorithms. In addition, sequences that have agility in the rest of the pulse parameters (mainly RF and PW) are problematic for clustering algorithms. It is desired to deal with jittered and staggered sequences, and at the same time to deal with agility in frequency and PW. Therefore, de-interleaving based on time-based algorithms alone or based on clustering algorithms alone is discarded. Moreover, algorithms that use non-conventional intra-pulse features (such as wavelet and resemblance coefficients) in de-interleaving are discarded in order to avoid the extra processing burden and integration complexity within EW system.

4.2 Approach

A combination of time-based algorithms and clustering algorithms will be used in order to provide a de-interleaving solution that can handle agility in time, in addition to the agility in RF and PW. The solution will include a time stage and a clustering stage. In the following sections, it is explained why the time-based algorithms available from literature are discarded from use in parallel with the clustering stage or even as a first stage of the desired de-interleaving solution. Based on that, the clustering stage will be the first stage of the algorithm. The alternatives of clustering algorithms (based on the de-interleaving literature) were reviewed. The desired clustering technique need to be unsupervised, not requiring prior information about clusters, immune to noise, and with low complexity. The review demonstrated that none of the existing clustering algorithms (regarding the reviewed de-interleaving literature) combines all these features. DBSCAN is an algorithm that has not been used before in the de-interleaving of radar pulses (to the best of our knowledge), but it satisfies all of our desired clustering features. Therefore, the DBSCAN algorithm is selected for the clustering stage.

In the case of RF agility of an emitter, the clustering stage will not be able to link the various identified clusters with their original emitter. In order to resolve the problem of emitter agility, the time parameter will be utilized. This parameter was chosen because it is the most reliable parameter among the rest of the parameters that have not been utilized in the clustering stage (of the proposed solution). The available time-based algorithms do not provide a solution with regard to the linking of related agile clusters. The use of time parameters in resolving emitter agility is illustrated in Section (4.4.3).

In fact, clustering and agility resolving are the major building blocks of the proposed algorithm.

However, the algorithm consists of other important blocks in addition to clustering and agility resolving, as illustrated in the system overview section (see Section (4.4)).

4.3 Time-based Stage

Time-based algorithms require a presorting stage in order to reduce the density of the environment. The AOA parameter is well known in the literature for being the most reliable sorting parameter, and it was thus used as a presorting parameter, as in [12]. The typical AOA accuracy for ELINT/ESM receiver is in the range of 5^o (this accuracy should not be confused with receivers that are used for ELINT-only, which can achieve much better accuracy; our discussion does not address this type of receivers). By examining the sorting that is based on AOA parameter, while taking the accuracy of the ELINT/ESM into consideration, it can be concluded that the AOA parameter is not a reliable parameter on its own for our ELINT/ESM receiver scenario. The probability that an emitter will have another overlapping emitter in the AOA domain is given by equation (4.16), which was derived in Section (4.4.3.2):

$$P(A) = \frac{N_{AOA}^k - \sum_{i=0}^{N_{AOA}-1} [N_{AOA} - \beta]^{(k-1)}}{(N_{AOA})^k} \tag{4.1}$$

where k is the number of emitters, N_{AOA} is the number of discrete AOA values, and β is the required separation in AOA measured in discrete points. The value of N_{AOA}, and β can be calculated by:

$$N_{AOA} = \frac{360^o}{\alpha} \tag{4.2}$$

$$\beta = \frac{2\Theta}{\alpha} + 1 \tag{4.3}$$

where α is the resolution of AOA, and Θ can be chosen to be three standard deviations of AOA error distribution.

By substituting the values: $\alpha = 5^o$, $\Theta = 15^o$, $k = 30$, then the probability in equation (4.1) becomes $\mathrm{P}(A) = 0.9485$. This probability of overlapping is very high, which makes separating the emitters based on their AOA unreliable.

To illustrate this, a simulation for randomly located emitters was performed using 30 emitters. The emitters are uniformly distributed in terms of their angle, and each emitter has Gaussian error distribution for its measured AOA parameters. The RMS error of the AOA parameter is 5^o. The histogram is plotted in order to provide an understanding of separability of the AOA parameter,

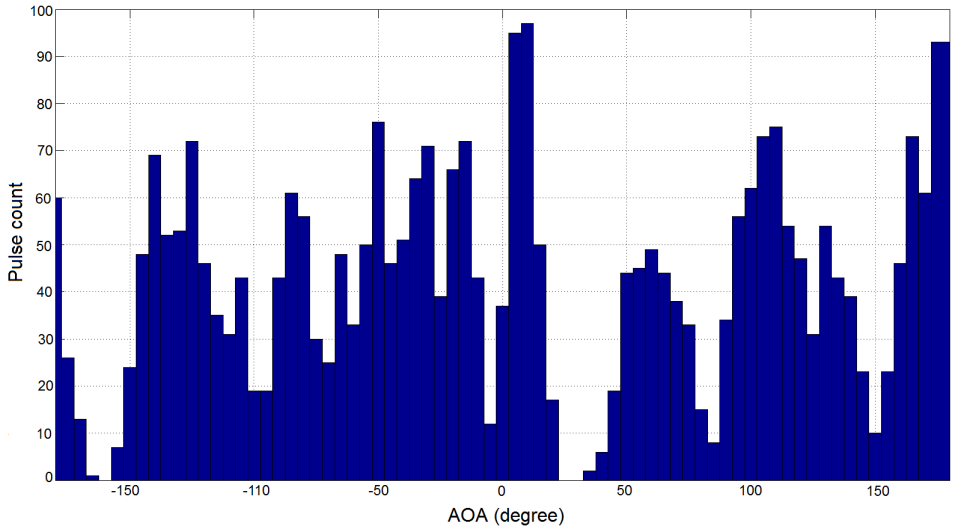


Figure 4.1: AOA histogram of randomly located emitters (30 emitters). The accuracy of AOA is 5^o.

as illustrated in Figure (4.2).

RF is the next reliable sorting parameter. The typical accuracy of RF is 1 MHz, which allows for better separation using RF compared to AOA. The histogram was plotted as shown in Figure (4.2) for 30 simulated clusters with random RF parameters. The RFs of clusters are uniformly distributed between 1000 MHz and 12000 MHz. The upper limit of 12000 MHz was chosen in this simulation because most of the emitters fall in this range. Obviously, better results are obtained if the upper limit is 18000 MHz. The measured RF parameters within a cluster have a Gaussian distribution with RMS error of 1 MHz. However, the problem of using RF as a presorting parameter is that the agile emitter pulses are fragmented into multiple groups. This fragmentation will cause a failure in available time-based algorithms. From the previous discussion, it can be concluded that the available time-based algorithms are not suitable for use as the first stage in the proposed de-interleaving solution.

4.4 System Overview

The proposed de-interleaving system consists of different stages. The major building blocks of the system includes: SNR Filter, PDW Filter, Time Segmentation, Clustering, Cluster's Continuity Detection, Classes Establishment, Agility Resolving, PRI Analysis, and Tracks Generation. The output of the system is managed in three different data structures: Clusters, Classes, and Tracks. The input to the system is the PDW sequence, which is provided by the EW receiver.

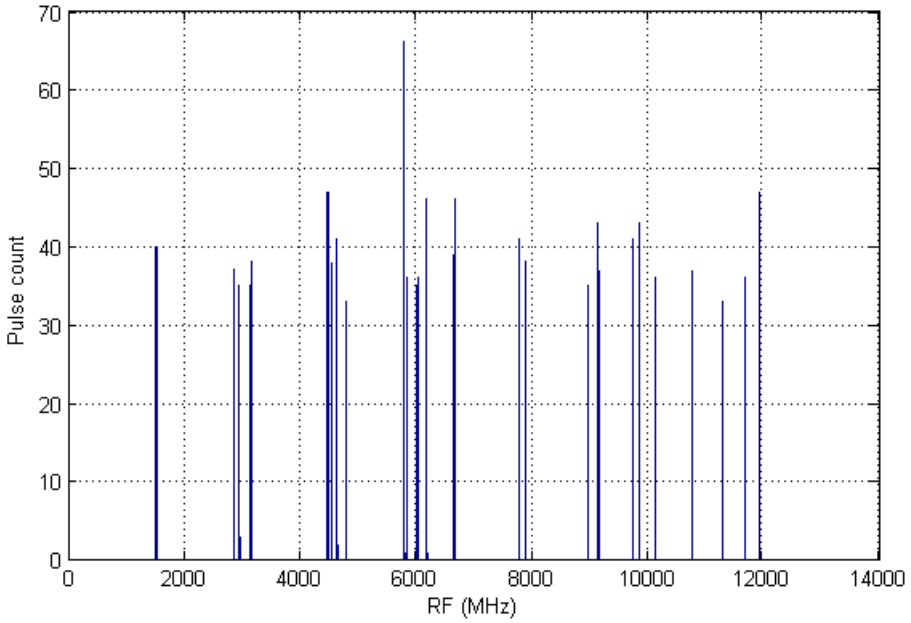


Figure 4.2: RF histogram of random emitters (30 emitters). The accuracy of RF is 1 MHz.

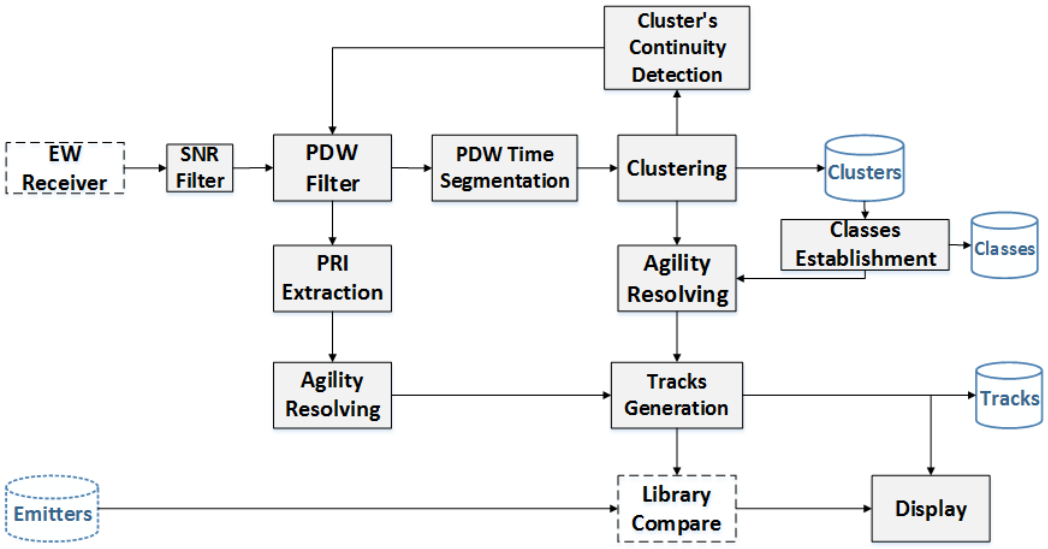


Figure 4.3: Proposed De-interleaving System.

4.4.1 Clustering Stage

Several algorithms for clustering radar parameters were found in the literature as presented in Chapter 3. The relevant algorithms for our scenario are SOFM, F-ART, SVC, and K-means. The problem with regard to SOFM is that it requires the number of clusters beforehand. Although Dai [21] used a CDBW validity index in order to provide this information for SOFM, the algorithm is still impractical because CDBW requires SOFM to be applied several times in order to determine the best number of clusters to be used. Furthermore, the CDBW validation index is sensitive to noise. K-means also need to know the number of clusters beforehand. SVC and F-ART do not require number of clusters in advance. However, SVC is computationally intensive, and it was thus used together with K-means in order to reduce the complexity as in [68]. For the same purpose, SVC (in another algorithm) is applied on several of short segments of data, and then the centroids of the resultant clusters are clustered again with SVC. Alternatively, F-ART is a good algorithm with regard to being able to preserve previously learned patterns, while also being able to learn new patterns at the same time; this is referred to as the stability-plasticity dilemma in the field of machine learning [93]. However, the problem of F-ART lies in its sensitivity to noise [94].

According to our requirements, we need to use a clustering algorithm that is unsupervised, does not require prior information about clusters, is immune to noise, and has low complexity. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was the clustering algorithm that satisfies these features.

4.4.1.1 DBSCAN Algorithm

Clustering is the process of grouping together PDWs that are similar in some way into distinctive clusters. A cluster is thus a collection of PDWs that are similar to other within their cluster and different from PDWs of other clusters. There are many categories of clustering techniques, which include "partitioning, hierarchical, grid, density, graph and model-based algorithms" [95]. DBSCAN is one of the density-based clustering techniques.

DBSCAN is regarded as the most used algorithm among the density-based clustering algorithms [96]. Clusters in DBSCAN are constructed based on two criteria, viz., density and connectivity. Clusters are formed from regions of data sets that have higher spatial density compared to the rest of the data set. One cluster is thus defined as a density-connected region.

DBSCAN performs clustering of data without requiring prior knowledge about the number of clusters. The algorithm requires only two parameters ($MinPts$, and Eps), which can be set according to the nature of the data. In addition, the algorithm is immune to outliers and can

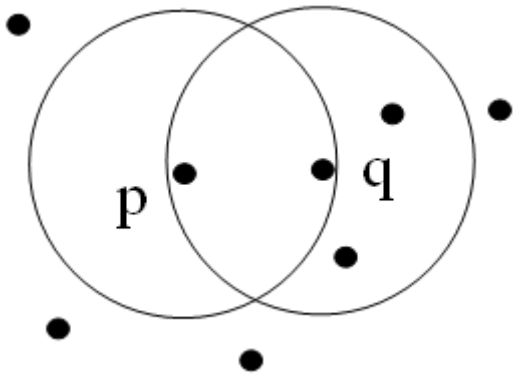


Figure 4.4: Directly density-reachable point. *Eps* is illustrated by circles, and MinPts are 4.

distinguish noise points. The algorithm is not sensitive to the order of the data, and moreover, the algorithm is simple.

The average run-time complexity of DBSCAN is $\mathrm{O}(\log(n))$ [42], but this is highly dependent on the implementation. The worst case run-time complexity is $\mathrm{O}(n^2)$, while the memory complexity is $\mathrm{O}(n^2)$, which is not good in general for long data sets. However, the run-time and memory complexities can be reduced according to the implementation being used.

In the following sections, various important definitions to DBSCAN are presented.

Eps-neighborhood points These are the points within a radius of *Eps*, where *Eps* is a predefined constant.

Core point This is a point that has a number of Eps-neighborhood points that are at least $MinPts$.

Directly density-reachable point If q is a core point, then p is directly density-reachable from q if p is an Eps-neighborhood of q. Figure (4.4) demonstrates the concept of the directly density-reachable point .

Density-reachable Point p is density-reachable from q if there is a chain of points p_1,p_2,...p_n, where $p_1 = q$, and $p_n = p$, and where p_{i+1} is directly density-reachable from p_i. Figure (4.5) illustrates the concept of the density-reachable point.

Density-connected If there are two points p and q, then these two points are density-connected, if there is a third point o such that both p and q are density reachable from o. Figure (4.6) illustrates the concept of density-connected points.

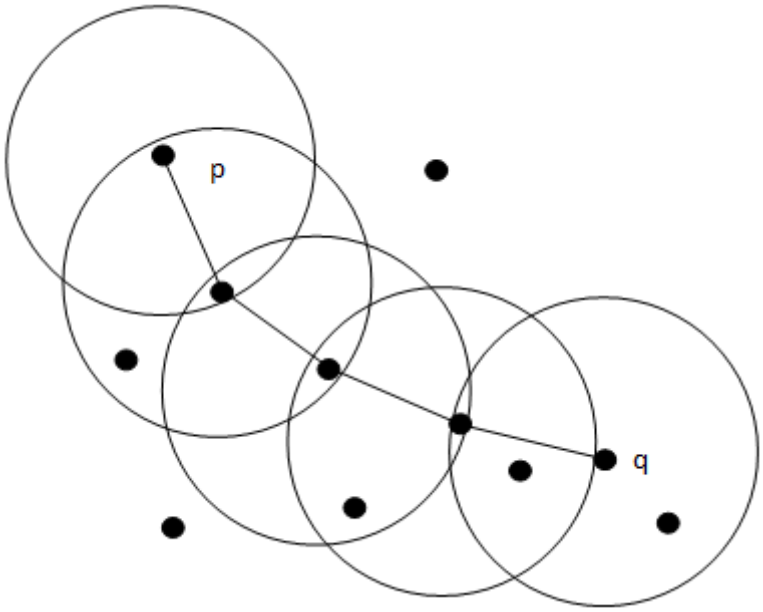


Figure 4.5: The point p is density-reachable from point q.

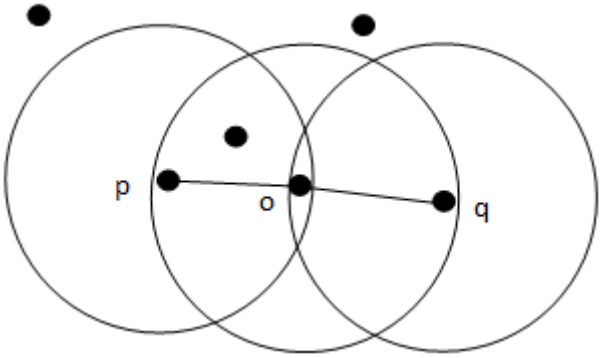


Figure 4.6: Points p and q are density-connected.

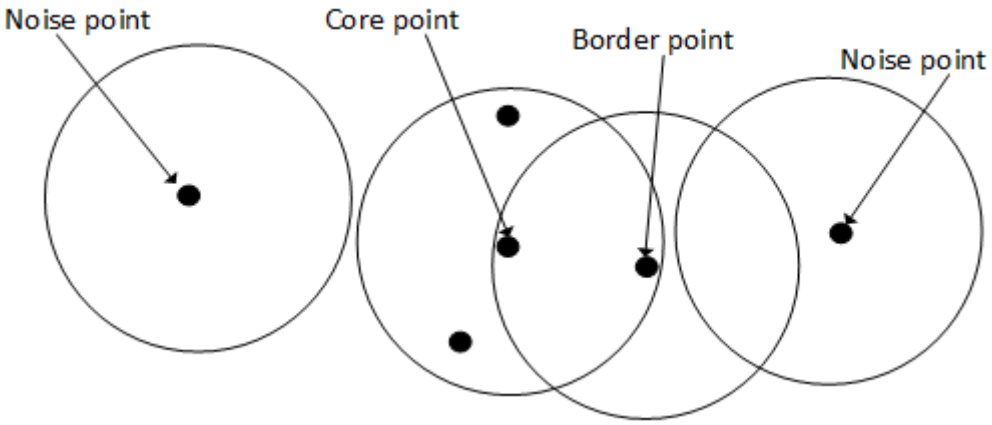


Figure 4.7: Illustration for core, border, and noise points.

Border point If a point p is within the Eps-neighborhood of a core point, but p itself is not a core point, then p is a border point.

Noise point This point that is neither a core point nor a border point. Figure (4.7) illustrates the noise point, and the border point.

Algorithm description DBSCAN starts with an arbitrary point that was not visited. Thereafter, the neighborhood points of this arbitrary point are found. If the point was found to be a core point (neighborhood points$>= MinPts$), then a cluster is formed, and all neighborhood points are added to the cluster. The arbitrary point is then marked as visited. The neighborhoods of the core points of the formed cluster will be added to the cluster. If the arbitrary point has neighborhood points $< MinPts$, then it will be marked as noise. All tested points will be marked as visited. If all members of formed cluster were visited, then new unvisited arbitrary point is chosen. All previous steps are repeated until all data points have been visited.

The flow diagram of the actual code used for clustering the PDWs is depicted in Figure (4.8) and Figure (4.9).

4.4.2 Pulse Sequence Segmentation

This block manages the size of the input buffer for the next clustering stage. The buffer input size should be limited to a maximum length, which can be obtained based on the performance of overall algorithm. Increasing the buffer size above a certain value can reduce the speed of the clustering stage and consequently the speed of the whole algorithm. Conversely, reducing the buffer size can also reduce the overall algorithm speed due to a reduction in the speed of the other algorithm

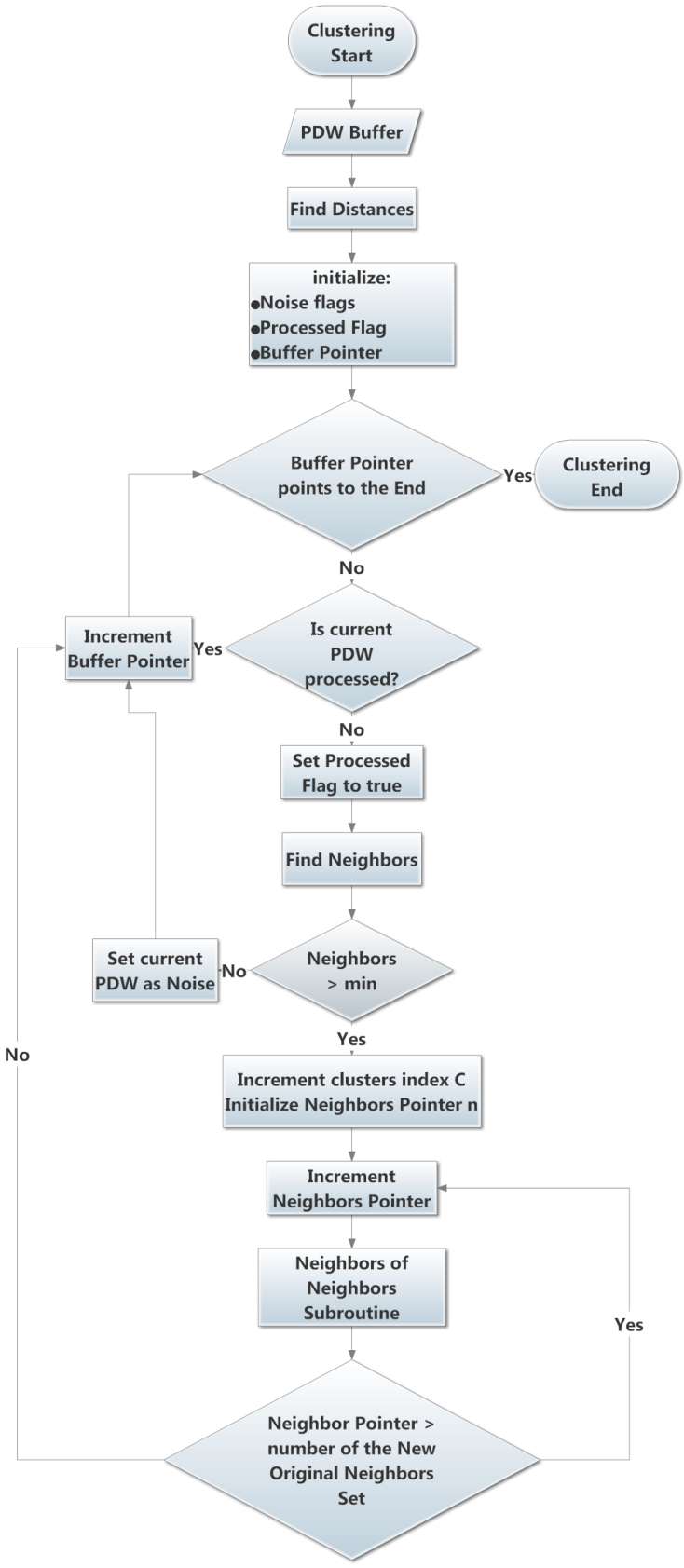


Figure 4.8: Flow chart diagram for clustering of PDWs based on DBSCAN. The Neighbors of neighbors subroutine block is illustrated in the next figure.

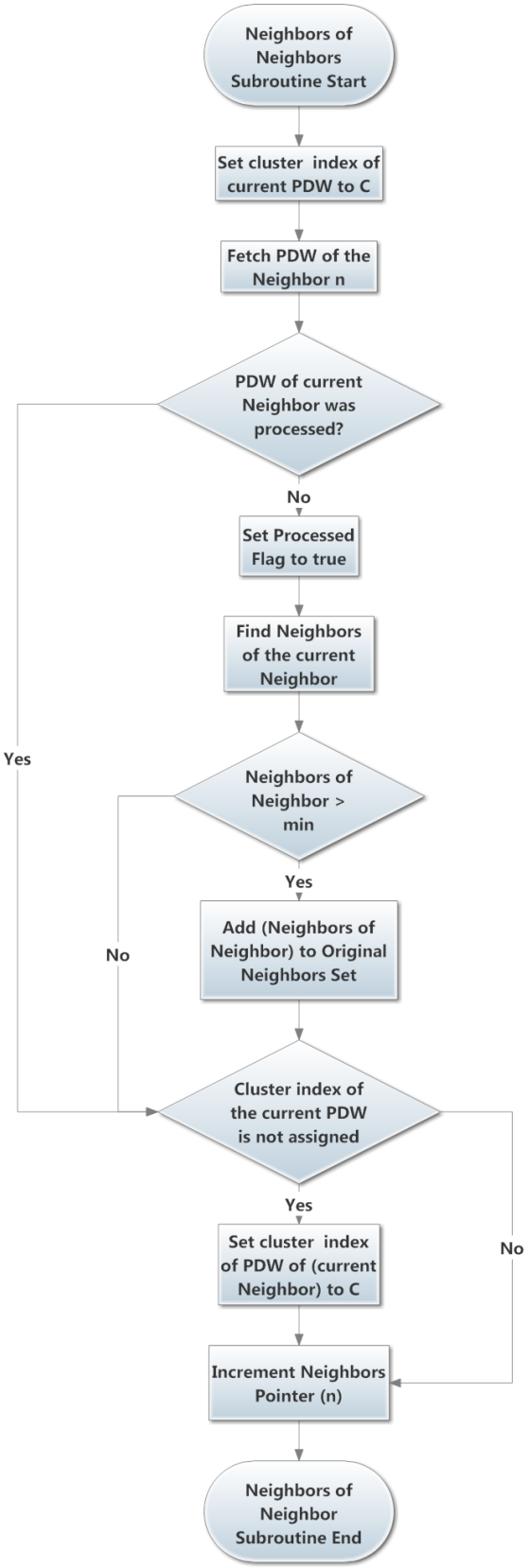


Figure 4.9: Neighbors of neighbors subroutine.

stages. In addition, the smaller buffer size could increase the chance of missing the detection of a cluster, particularly when the emitter has a few pulses that are fragmented between two buffers.

Two cases need to be considered when talking about buffer management: the continuous pulse stream environment and the non-continuous pulse stream environment. In the case of the former, it is desired to reduce the number of the interruptions (that are made by the buffer) of the pulse stream. This can be achieved by maximizing the size of the buffer. On other hand, the time that is required in order to fill the buffer should be minimized in order to speed up the algorithm. The size of the buffer should be also reasonable for the computational complexity of clustering algorithm.

Assuming the maximum acceptable time to fill the buffer is 1 second, then, if the lowest average continuous pulse rate is 1 kpps (kilo pulse per second), then the maximum buffer size is 1000 pulses. If the extreme pulse density of the environment is 1 Mpps (as found in most literature), then the time that is required in order to fill a buffer of length of 1000 pulses is 1 ms. Let us assume the chosen buffer size is 1000 pulses.

Now let us consider the case of noncontinuous pulse stream environment with the chosen buffer size above. Let us consider the case when only one scanning radar is in the environment, and when 6 pulses on target are expected every 12 seconds (average of 0.5 pps). In this case, the buffer needs 2000 s (33.3 min) in order to be filled, which is prohibitively high. Silence period detection is introduced in order to address this problem.

Silence Period Detection This function declares a period of silence, when no pulses are received at the EW receiver input, for a sufficiently long time. In order to determine this time, assume the maximum expected radar range is 500 Km. This corresponds to a PRI of 3.3 ms. Assume the number of pulses in the main beam is 10. Assume all of these pulses were received with a possible loss of 1 or 2 consecutive pulses. The silence period resulting from 2 consecutive missing pulses is $(2+1)\,3.3 = 10\,ms$. Therefore, if the time to the next pulse exceeds 10 ms, then a silence period is declared.

The noise in the EW receiver causes false pulse detections and consequently the generation of incorrect PDWs (PDWs noise). In spite of the fact that the PDWs noise will be always present in the EW receiver output, this presence should not be in a continuous pattern. Having continuous (highly frequent) PDWs noise in EW receiver output implies a high probability of false alarm, which is acceptable only in specific applications. If this type of continuous noise exists, then silence detection will still be functional beneath a certain continuous noise rate threshold. Consider the case when one scanning radar is in the environment, while it is masked by a continuous noise stream. The noise stream will constitute continuity in the whole pulse stream, and it prevents occurrence of

silence periods. Consequently, the detection of the scanning radar cluster will be severely delayed. This can happen when the time between *continuous* noise pulses is less than the value of the silence period threshold (10 ms). Hence, if the continuous noise rate is less than 100 pps, then the silence events will not be masked by noise. For the case of *non-continuous* noise, this type of noise will not cause a problem for silence detection, regardless of the time between noise pulses.

4.4.3 Agility Resolving

RF agile emitters have more than one cluster (for each emitter). It is necessary to find a possible relationship between clusters that belong to an RF agile emitter in order to link them to their emitter. In our scenario, finding such a relationship should only rely on standard PDW parameters. The parameters of AOA and RF were already used in the clustering stage. PW and amplitude are not as reliable as TOA parameter, however, and hence TOA is selected in order to explore the agility relationship.

The discussion focuses on the agility that happen within a short time frame (few milliseconds). This type of agility includes pulse-to-pulse, and burst-to-burst (short bursts) agility. Agility resolving is done in two stages: firstly, the agility hypothesis generation stage and secondly, the agility hypothesis test stage, both of which are explained in the following subsections.

4.4.3.1 Agility Hypothesis Generation

As the number of clusters under test increases, the problem of resolving agility becomes more complex. The complexity of the agility test can be reduced if the test is limited to clusters that have a potential agility relationship. Two conditions must be met in order to consider a group of clusters to have a potential agility relationship. Firstly, all clusters must have RF centers that are within 10% RF-distance from the RF-center of the cluster under test. Secondly, these clusters must also have an AOA-distance between their AOA-centers that is less than 5^o. In the hypothesis stage, one cluster can be part of multiple agile groups.

4.4.3.2 Agility Hypothesis Test

As mentioned before, the time parameter will be used in order to perform the agility test. Under conditions of PDW fragmentation (due to agility), the known time-based algorithms are not useful. It was thus decided to rely on the time of the cluster instead of the known TOA parameter. Cluster-time (the center time of the cluster) is obtained from TOA and it gives indication about the time of scanning.

The potential agility hypothesis stage provides sets of clusters that may have an RF agility relationship. Sometimes one cluster or more, or even none of the clusters in a provided hypothesis set, are related to an agile emitter. Therefore, each set of clusters will be tested for its agility relationship.

A radar with RF agility will produce a number of clusters that are more than one, which will be detected by DBSCAN. When RF agility occurs within the same radar beam, i.e. on a pulse to pulse basis, then the time differences between the reception of these clusters are relatively very small. However, in some cases, time differences between clusters may be relatively small, although there are no real agility relationships. If this case happen for one scan, it will be very difficult to repeat it for the next scans.

The environment contains several scanning radars. In rotational scanning, each radar has a scan period that correspond to one antenna revolution period. In addition, each scanning radar will have a revolution-phase with respect to the EW-receiver. Two different scanning radars will *always* (for each scan) point to EW-receiver simultaneously (time overlap), but only if their scanning periods are equal, and only if both of them have a synchronous revolution-phase with respect to the EW-receiver. Synchronous phase means that the revolution phase of the radar is 180^o out-of-AOA direction, while AOA is measured with respect to the EW-receiver. The geometry for the synchronous revolution-phase is illustrated in Figure (4.10).

In fact, in order to have a false positive-test-result in repeatedly, the following conditions must be met;

- Identical revolution periods,
- Synchronous revolution phase,
- RFs of overlapping scans are within 10% separation,
- AOAs of overlapping scans are within 5^o,

Let us refer to these conditions by events, A, F, R, and ψ. The probabilities of these events are $P(A)$,$P(F)$,$P(R)$, and $P(\psi)$ respectively.

For the sake of argument, let us examine these conditions for two emitters. Assume that AOA is uniformly distributed on the interval $[0, 2\pi)$, and that RF is uniformly distributed over the interval $[f_{min}, f_{max}]$ MHz. The revolution phase is uniformly distributed over the interval $[0, 2\pi)$.

The resolution of AOA is α, and the discrete AOA interval is $[0, N_{AOA} - 1]$, while;

$$N_{AOA} = \frac{360^o}{\alpha} \tag{4.4}$$

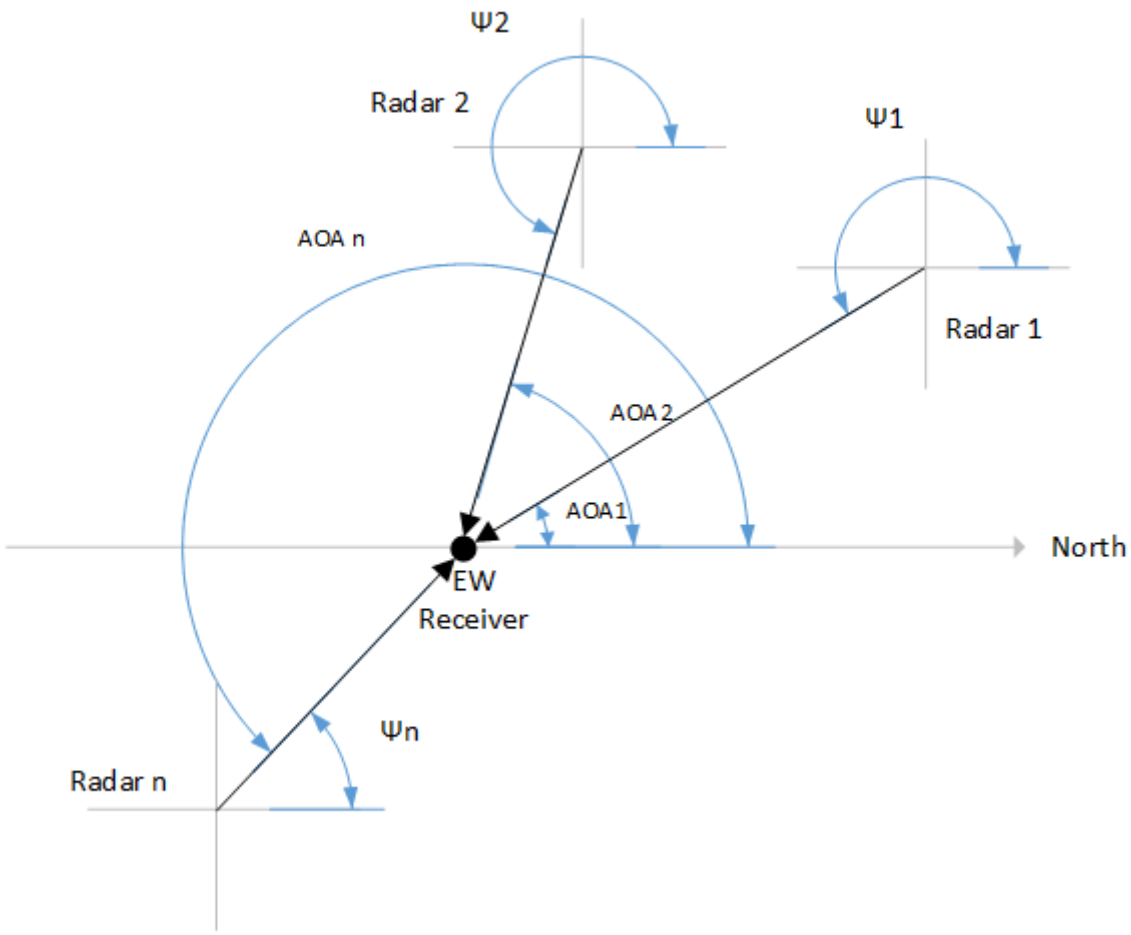


Figure 4.10: Geometry of emitters and EW receiver.

The resolution of RF is γ, and the discrete RF interval is $[F_{min},F_{max}]$.

Let us first discuss the probability of receiving two emitters, while one of them is within 10% RF distant from the other. The number of possible discrete RF values in EW receiver is given by:

$$N_f = \frac{F_{max} - F_{min}}{\gamma} + 1 \tag{4.5}$$

while N_f is an integer.

The RF domain of EW receiver can be described by:

$$RF\left(i\right) = \gamma i + F_{min}, i \in [0, N_f - 1] \tag{4.6}$$

According to the 10% RF separation condition, the second emitter can have a frequency in the range $\pm 0.1RF\left(i\right)$.

$$P\left(F\right) = \frac{\sum_{i=0}^{N_f-1}\left[0.2\left(F_{min} + \gamma \cdot (i)\right) + 1\right]}{\left(N_f\right)^2} \tag{4.7}$$

$$P\left(F\right) = \frac{\left(1/10\right) N_f \left(N_f + \frac{2F_{min}}{\gamma} + 10\right)}{\left(N_f\right)^2} = \frac{\left(N_f + \frac{2F_{min}}{\gamma} + 10\right)}{10N_f} \tag{4.8}$$

Equation (4.7) is valid for two emitters only. In order to generalize the result in order to include k emitters, let us first re-write equation (4.7) by adding N_f^2 to the numerator and subtracting it.

$$P\left(F\right) = \frac{N_f^2 - N_f^2 + \sum_{i=0}^{N_f-1}\left[0.2\left(F_{min} + \gamma \cdot i\right) + 1\right]}{\left(N_f\right)^2} \tag{4.9}$$

$$P\left(F\right) = \frac{N_f^2 - \sum_{i=0}^{N_f-1} N_f + \sum_{i=0}^{N_f-1}\left[0.2\left(F_{min} + \gamma \cdot i\right) + 1\right]}{\left(N_f\right)^2} \tag{4.10}$$

$$P\left(F\right) = \frac{N_f^2 - \sum_{i=0}^{N_f-1}\left[N_f - 0.2\left(F_{min} + \gamma \cdot i\right) + 1\right]}{\left(N_f\right)^2} \tag{4.11}$$

For k emitters we can prove that:

$$P\left(F\right) = \frac{N_f^k - \sum_{i=0}^{N_f-1}\left[N_f - 0.2\left(F_{min} + \gamma \cdot i\right) + 1\right]^{(k-1)}}{\left(N_f\right)^k} \tag{4.12}$$

The probability of receiving two emitters, while they are separated by less than Θ^o is given by:

$$p\left(A\right) = \frac{\beta \cdot N_{AOA}}{N_{AOA}^2} = \frac{\beta \cdot \alpha}{360} \tag{4.13}$$

while β is the separation measured in discrete points, and given by:

$$\beta = \frac{2\Theta}{\alpha} + 1 \tag{4.14}$$

Therefore, the probability of receiving two emitters, while they are separated by less than Θ^o, becomes:

$$P(A) = \frac{\alpha}{360}\left(\frac{2\Theta}{\alpha} + 1\right) \tag{4.15}$$

By applying same technique in equation (4.7) to equation (4.13):

$$P(A) = \frac{N_{AOA}^k - \sum_{i=0}^{N_{AOA}-1}\left[N_{AOA} - \beta\right]^{(k-1)}}{\left(N_{AOA}\right)^k} \tag{4.16}$$

Now, let us discuss the revolution phase of emitters, which is a continuous space variable $\psi \in [0, 2\pi)$. Let us assume that the revolution phases of two emitters are identical, if $\nabla\psi \leq \rho$. In same manner to equation (4.16), $P(\psi)$ can be found such as:

$$P(\psi) = \frac{(2\pi)^k - \int_0^{2\pi}(2\pi - \rho)^{k-1}\, d\psi}{(2\pi)^k} \tag{4.17}$$

$$P(\psi) = 1 - \frac{(2\pi - \rho)^{k-1}}{(2\pi)^{k-1}} \tag{4.18}$$

$$P(\psi) = 1 - \left(1 - \frac{\rho}{2\pi}\right)^{k-1} \tag{4.19}$$

Now, given that two emitters have close frequencies, then they usually perform the same radar function, i.e., early warning, acquisition, navigation, etc. Early warning radars for instance, are usually located in the frequency range (1000-4000) MHz because of various design considerations. The PRFs of these radars are relatively small, and therefore their revolution period is long (to allow enough number of pulses to reach the target). In comparison, navigation radars, which are usually located around 9000 MHz, have a smaller revolution period and a higher PRF. Therefore, let us assume that, in the case where two emitters have close frequencies, the chance that both of the emitters have same scan period is $P(R_2 \mid F_2)$, i.e. R and F are dependent. Let us assume $R \in [1, 12]$ seconds, which corresponds to scan frequency $\Omega \in \left[\frac{2\pi}{12}, 2\pi\right] \frac{rad}{sec}$.

$$P(\Omega) \leq P(\Omega \mid F) < 1 \tag{4.20}$$

Let us consider that two different scan frequencies are identical, when the difference in the scan phase after 5 scan revolutions is less than 1.5^o. If the difference in the scan frequency is ε, then;

$$\Omega_1 t = \theta_1 \tag{4.21}$$

$$\Omega_2 t = (1 + |\varepsilon|)\, \Omega_1 t = \theta_2 \tag{4.22}$$

$$\nabla\theta = |\varepsilon| \Omega_1 t \leq \frac{1.5}{360} \cdot 2\pi \tag{4.23}$$

$$|\varepsilon| \cdot \frac{2\pi}{T} \cdot 5T \leq \frac{1.5}{360} \cdot 2\pi \tag{4.24}$$

$$|\varepsilon| \leq \frac{1.5}{(360)\,(5)} \tag{4.25}$$

$$P(\Omega) = \frac{\left(\frac{11}{6}\pi\right)^k - \int_{\frac{2\pi}{12}}^{2\pi} \left(\frac{11}{6}\pi - 2\varepsilon\right)^{k-1} d\psi}{\left(\frac{11}{6}\pi\right)^k} \tag{4.26}$$

$$P(\Omega) = 1 - \left(1 - \frac{12\varepsilon}{11\pi}\right)^{k-1} \tag{4.27}$$

Now, the probability of repeatedly obtaining a false positive test result for a given emitter can be given by:

$$P(E) = P(F \cap \Omega \cap A \cap \psi) \tag{4.28}$$

$$P(F \cap R) = P(F) \cdot P(\Omega \mid F) \tag{4.29}$$

$$P(F \cap \Omega \cap A \cap K) = P(F) \cdot P(\Omega \mid F) \cdot P(A) \cdot P(\psi) \tag{4.30}$$

Assuming $\gamma = 1MHz$, $\alpha = 5^o$, $\Theta = 15^o$, $\rho = 1.5^o$, and $k = 30$, then;

$P(F) = 0.8834$, $P(A) = 0.9485$, $P(\psi) = 0.2155$, and $0.0239 \leq P(\Omega \mid F) < 1$.

Hence, the probability of repeatedly obtaining a false positive test result is:

$$0.0043 \leq P(E) < 0.1806$$

Cluster time segmentation It is not common to have multiple scans of the same radar in one buffer, in the normal settings. However, if this situation were to happen, it could lead to confusion when comparing the times of the different clusters. Therefore, for each resultant cluster, it is required to know if the cluster consists of multiple scans occurring within the buffer.

Cluster timing test As mentioned before, the times of the clusters under test will be compared. If clusters contain more than one time (time segments), then all the time segments will be compared. If time differences are more than a certain threshold, then clusters are considered to have positive agility relationship, otherwise, the test result is negative. This stage will thus establish the pairs of agile classes. Only some information about the agility relationships is obtained during this stage. Scores from successive processing are maintained and updated for each agile class pair.

The test is done repeatedly as mentioned before.

Clusters Linking In the clustering stage, DBSCAN is performed sequentially on batches of incoming pulses. The relationship map between all the detected classes will be established. This relationship map will be updated as new PDWs batches are processed by DBSCAN. As mentioned before, the agility test is done repeatedly, which removes the possible noisy results of the agility test. The weights of the links between the pairs of detected agile classes are processed in order to omit noisy relationships. Once this process is completed, pairs of agile classes are further processed in order to combine related agile pairs. This stage produce groups of connected classes, where each group consists of one or more classes. Each group represents a different radar.

4.4.4 Cluster's Continuity Detection

Emitters with pulse streams that are continuously present in the buffer can cause a problem for the agility resolving stage. In addition, these emitters will add more load on the processing side of the algorithm. Hence, it is necessary to detect the presence of clusters that have continuous pulses. Once these clusters are identified, then the corresponding PDWs will be filtered out from the upcoming buffers. Moreover, these clusters with "Continuous Sequence" identification will be handed over to different agility test stage.

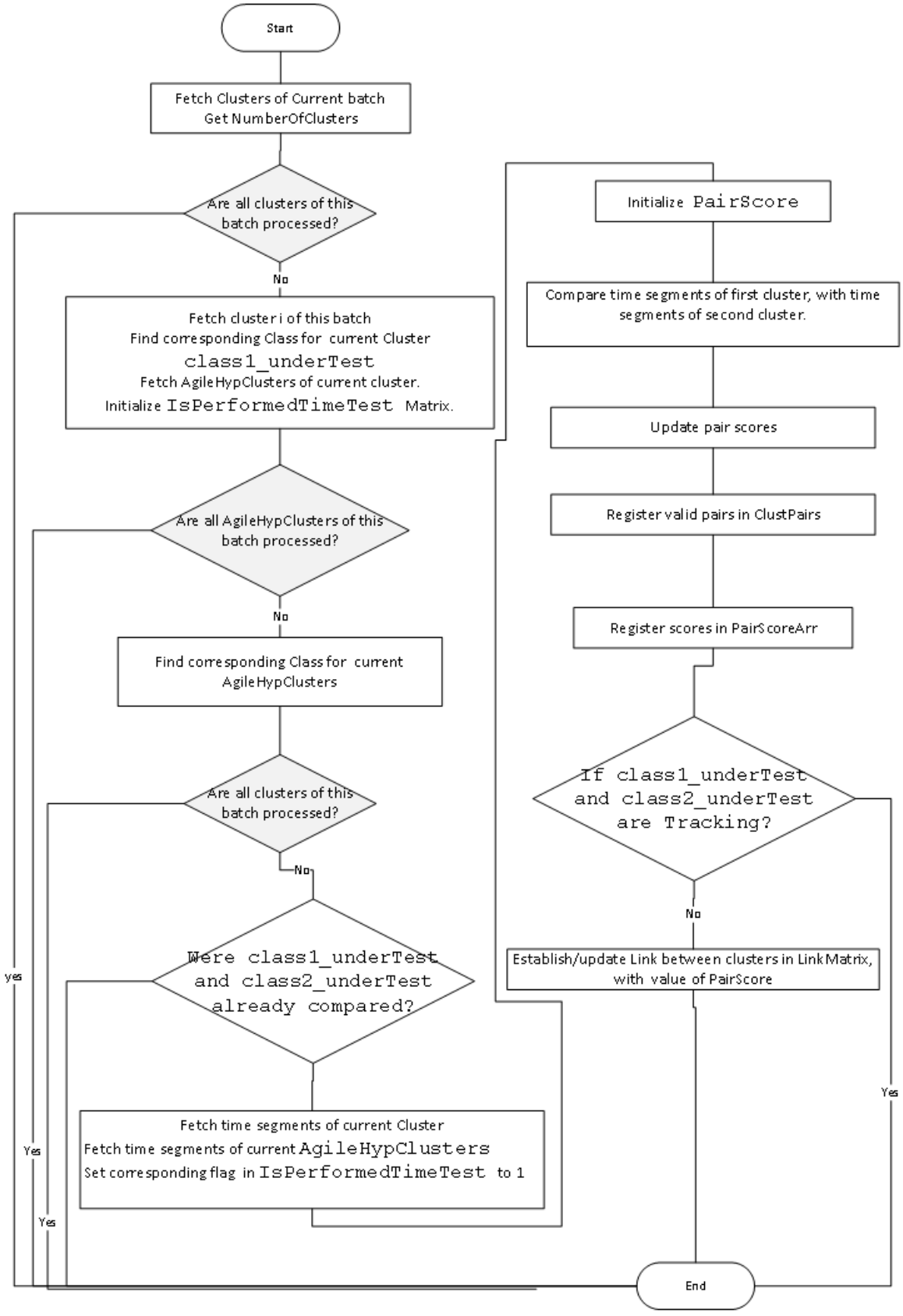


Figure 4.11: Linking agile clusters - First stage.

4.4.5 PRI Analysis

It is desired to use PRI analysis of clusters in order to resolve the agility of continuous-pulse-stream emitters. When an emitter is frequency agile, then its pulses are fragmented into several clusters. Therefore, PRI values of these clusters are not same as the PRI of the originating emitter.

In the presented PRI analysis, the intention is to gather information that assists in linking related clusters. Hence, obtaining the PRI parameter of the original emitter is not the main interest of this analysis.

In this section, two PRI analysis method are proposed. The first utilizes Least Squares concept, while the second relies on the difference of the TOA parameter.

4.4.5.1 Utilizing Least Squares Concept

This section introduces a characterization (modeling) for a sequence of a cluster that belongs to a frequency agile emitter. It thereafter presents the least squares concept that is used to estimate the PRI of the clusters. After that, the analysis presents the least squares estimation method utilizing presented cluster sequence model. And, finally, it shows the feasibility of using the PRI analysis results (using the least squares concept) in linking related clusters.

In the case of frequency agile emitters, the emitter has stable PRI from pulse to pulse, but its frequency is not fixed. In fact, the change in frequency from pulse to pulse can follow different patterns. Each cluster that results from a given agile emitter can have one or more PRI values, depending on the RF-sequence pattern used by that emitter. To illustrate this, let us take an emitter that has N different frequencies ($F_1, F_2, F_3, ...F_N$). The simplest sequence pattern is $\{F_1, F_2, F_3, ...F_N, F_1, F_2, F_3, ...F_N, ...\}$. Obviously, this emitter produces N clusters. If this emitter has $PRI = T$, then the pulse sequence for each resultant cluster (C_n) is given by:

$C_1 = \{0, NT, 2NT, ..\}$,

$C_2 = \{T, (N+1)\,T, (2N+1)\,T, ..\}$,

$C_3 = \{2T, (N+2)\,T, (2N+2)\,T, ..\}$, and

$C_n = \{(n-1)\,T, (N+n-1)\,T, (2N+n-1)\,T, ..\}$.

The PRI for each cluster is NT. The previous pattern produced clusters, each of which has one PRI value. The following example relate to a pattern that produces clusters, where each has two different PRI values. The pattern is as follows;

$F_1, F_1, F_2, F_2, F_3, F_3, ..., F_N, F_N$,

where,

$C_1 = \{0, T, (N+1)\,T, (N+2)\,T, (2N)\,T, ..\}$,

$C_n = \{2(n-1)T, (2n-1)T, (2n-1+N)T, (2n+N)T, (2n+N+1)T, ..\}$, etc.

Each cluster in this pattern has two different PRI intervals, viz., T, and NT. In another example, different clusters can also take different combinations of PRI values, as in the pattern

$F_1, F_1, F_1, F_2, F_3, ..., F_N$, for instance. This pattern has one cluster with two PRI intervals, whereas the rest of the clusters have one PRI interval. There is a variety of possible sequence patterns (P), which can be characterized differently (C_n).

In fact, the reason behind finding PRI of clusters is not in PRI value itself, but it is rather to find a mean in order to link related agile clusters. Therefore, the least square method will be used to estimate the resultant PRI value of clusters. The method of least squares can estimate the best fitting straight line for a given set of data points. Each cluster has a TOA sequence, which represents the given set of data points. Consider each TOA sequence is described by:

$$t(n) = T \cdot n + t_0 \tag{4.31}$$

where: n is pulse serial number within a cluster, such that $n = [0, 1, 2, .., N-1]$. N is the number of pulses. $t(n)$ is TOA value of pulse n, T is PRI value of cluster, and t_0 is time displacement. In general, for a given sample set $\{(x_1, y_1), ..(x_m, y_m)\}$, the parameters of the best matching line;

$$y = b\,x + a \tag{4.32}$$

can be estimated by [97]:

$$\begin{bmatrix} m & \sum x_j \\ \sum x_j & \sum x_j^2 \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sum y_j \\ \sum x_j y_j \end{bmatrix} \tag{4.33}$$

Hence the parameters in equation (4.31) are estimated by;

$$\begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} t_0 \\ T \end{bmatrix} = \begin{bmatrix} \sum_{n=1}^{N} t(n) \\ \sum_{n=1}^{N} n \cdot t(n) \end{bmatrix} \tag{4.34}$$

where;

$$s_{11} = N \tag{4.35}$$

$$s_{12} = s_{21} = \sum_{n=0}^{N-1} n \tag{4.36}$$

$$s_{22} = \sum_{n=1}^{N-1} n^2 \tag{4.37}$$

.

Knowing that, $n = [0, 1, .., N-1]$, then equation (4.36) and equation (4.37) can be evaluated as follows;

$$\sum_{n=0}^{N-1} n = \frac{1}{2} N (N+1) \tag{4.38}$$

$$\sum_{n=0}^{N-1} n^2 = \frac{1}{6} N (N-1) (2N-1) \tag{4.39}$$

Equation (4.34) can be rearranged as shown in equation (4.40), while the elements[1] of the matrix S are s_{11}, s_{12}, s_{21}, and s_{22}.

$$\begin{bmatrix} t_0 \\ T \end{bmatrix} = S^{-1} \cdot \begin{bmatrix} \sum_{n=1}^{N} t(n) \\ \sum_{n=1}^{N} n \cdot t(n) \end{bmatrix} \tag{4.40}$$

$$S^{-1} = \frac{1}{s_{11} s_{22} - s_{12}^2} \begin{bmatrix} s_{22} & -s_{12} \\ -s_{12} & s_{11} \end{bmatrix} \tag{4.41}$$

$$T = \frac{1}{s_{11} s_{22} - s_{12}^2} \left[s_{11} \sum_{n=0}^{N-1} n \cdot t(n) - s_{12} \sum_{n=0}^{N-1} t(n) \right] \tag{4.42}$$

Assume the frequency values are F_h, while, $h \in [1, H]$, and H is the number of frequencies. Therefore, we have H clusters C_h. Assume the pattern P describes RF sequence of an emitter that consists of multiple clusters. The pattern P consists of K elements, while $K \geq H$.

Now, the pattern P will be represented by H different sub-pattern P_h. The sub-pattern P_h is a set that consist of all indexes of the frequency F_h within the pattern P. All sub-patterns P_h are mutually exclusive.[2]

$$P_a \cap P_b = \Phi \quad \forall\, a \neq b, \quad a, b \in [1, H] \tag{4.43}$$

To illustrate that let us give an example for an emitter that has three different frequencies. The RF sequence pattern for this example can be given by;

[1] The elements s_{11}, s_{12}, s_{21}, and s_{22} are not related to the known s-parameters that are used in the field of radio frequency circuit design.

[2] The symbol Φ refers to the empty set.

$$P = \{F_1, F_2, F_1, F_1, F_2, F_3\}$$

The sub-patterns for this example are;

$P_1 = \{0, 2, 3\}$, $P_2 = \{1, 4\}$, *and* $P_3 = \{5\}$.

Now each cluster is represented by one sub-pattern. The time-sequence for a cluster can be obtained from its RF sub-pattern. If the sub-pattern (P_h) consists of J_h elements, then the time-sequence of the cluster can be represented as superposition of J_h different time-sequence C_{hj}. Let us call C_{hj} as time sub-sequence. Each time sub-sequence is given by:

$$C_{hj}(i) = K \cdot T \cdot (i-1) + T \cdot P_{hj} + t_o \quad , i \in [1, floor\left(\frac{t_{max} - t_o}{KT}\right) = I_h] \tag{4.44}$$

While, $j \in [1, J_h]$, P_{hj} is the value of the element with index j of the sub-pattern P_h.

Time sequence of the emitter is $t(n)$, while $t_h(n)$ is time sequence of cluster. C_{hj} is time sub-sequence (of cluster). Hence:

$$C_{hj} \subseteq t_h(n) \tag{4.45}$$

$$t_h(n) \subseteq t(n) \tag{4.46}$$

$$T_h = \frac{1}{s_{11h}s_{22h} - s_{12h}^2}\left[s_{11h}\sum_{n=0}^{s_{11h}-1} n \cdot t_h(n) - s_{12h}\sum_{n=0}^{s_{11h}-1} t_h(n)\right] \tag{4.47}$$

The parameter s_{11h} represents the number of pulses that belong to the sub-pattern P_h. Therefore, s_{11h} was used in the summation limits of equation (4.47), which is similar to equation (4.42). Suppose the number of elements in RF sub-pattern P_h is J_h. And assume the number of frames (the frame represents single RF-sequence pattern of the emitter) that constitute the time sequence of the emitter is I. Hence, the s_{11h} based on equation (4.34) is given by:

$$s_{11h} = J_h I \tag{4.48}$$

The indexes of pulses within the cluster are n_h. While:

$$n_h = \{0, 1, ..., (J_h I - 1)\} \tag{4.49}$$

Hence;

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \sum_{n=0}^{s_{11h}-1} (n_{h1} \cup n_{hj} \cup ... \cup n_{hJ_h})^2 \tag{4.50}$$

where, n_{hj} is the sample indexes of time sub-sequence C_{hj} with respect to the time-sequence of the cluster h. Hence;

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \frac{1}{6} J_h I \left(J_h I - 1\right) \left(2J_h I - 1\right) \tag{4.51}$$

or

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \frac{1}{6} s_{11h} \left(s_{11h} - 1\right) \left(2s_{11h} - 1\right) \tag{4.52}$$

$$y_1 = \sum_{n=0}^{s_{11h}-1} t_h (n) = \sum_{n=0}^{s_{11h}-1} (C_{h1} \cup C_{hj} \cup ... \cup C_{hJ_h}) \tag{4.53}$$

$$\sum_{n=0}^{s_{11h}-1} (C_{h1} \cup C_{hj} \cup ... \cup C_{hJ_h}) = \sum_{i=0}^{I-1} C_{h1}(i) + ... \sum_{i=0}^{I-1} C_{hj}(i) + .. + ... \sum_{i=0}^{I-1} C_{hJ_h}(i) \tag{4.54}$$

$$\sum_{n=0}^{s_{11h}-1} t_h (n) = \sum_{j=1}^{J_h} \sum_{i=0}^{I-1} C_{hj} \tag{4.55}$$

$$\sum_{i=0}^{I-1} C_{hj} (i) = \sum_{i=0}^{I-1} K \cdot T \cdot (i-1) + P_{hj} + t_o \tag{4.56}$$

$$y_1 = T \left(\frac{1}{2} I K J_h^2 \left(I J_h - 3\right) + A \upsilon_h \right) \tag{4.57}$$

For complete derivation of y_1 refer to Appendix (A).

$$y_2 = \sum_{n=0}^{s_{11h}-1} n_h \cdot t_h (n) \tag{4.58}$$

$$y_2 = \sum_{j=1}^{J_h} \sum_{n=0}^{s_{11h}-1} n_{hj} \cdot C_{hj} (n) \tag{4.59}$$

$$n_{hj} (i) = J_h \cdot (i-1) + j \tag{4.60}$$

$$\sum_{j=1}^{J_h}\sum_{n=0}^{I-1} n_{hj}(n)\cdot C_{hj}(n) \tag{4.61}$$

Note that, the size of sub-sequences, their number of elements, and their elements values are unknown. Hence, the terms $\sum_{j=1}^{J_h} P_{hj}$, and $\sum_{j=1}^{J_h} jP_{hj}$ cannot explicitly evaluated. In order to give an example for calculating these two terms let us take this case:

$P = \{F_2, F_1, F_1, F_1, F_2, F_3\}$. Hence:

$P_1 = \{1, 2, 3\}$

$P_2 = \{0, 4\}$

$P_3 = \{5\}$

$\sum_{j=1}^{J_h=3} P_{1j} = 1 + 2 + 3$

$\sum_{j=1}^{J_h=2} P_{2j} = 0 + 4$

$$y_2 = -\frac{3}{4}IKTJ_h^2+\frac{17}{12}IKTJ_h^3+\frac{1}{4}I^2KTJ_h^3-\frac{5}{4}I^2KTJ_h^4+\frac{1}{3}I^3KTJ_h^5+ITJ_h\rho_h-\frac{3}{2}ITJ_h^2\upsilon_h+\frac{1}{2}I^2TJ_h^3\upsilon_h \tag{4.62}$$

where, $\upsilon_h = \sum_{j=1}^{J_h} P_{hj}$, and $\rho_h = \sum_{j=1}^{J_h} jP_{hj}$.

For complete derivation of y_2 refer to Appendix (A).

$$s_{12h} = \sum_{n=0}^{s_{11h}-1} n = \frac{1}{2}J_hI(J_hI-1) \tag{4.63}$$

The value of T_h is given by:

$$T_h = \frac{1}{s_{11h}s_{22h}-s_{12h}^2}\left[s_{11h}\sum_{n=0}^{s_{11h}-1} n\cdot t_h(n) - s_{12h}\sum_{n=0}^{s_{11h}-1} t_h(n)\right] \tag{4.64}$$

$$T_h = KT\left(\frac{17}{I^2}+\frac{15}{I}-\frac{18}{I^2J_h}-3J_h-\frac{15J_h}{I}+4J_h^2\right)+T\left(\frac{12\rho_h}{I^2J_h^3}+\frac{(6-18J_h)\upsilon_h}{I^2J_h^3}\right) \tag{4.65}$$

$$T_h = KT\left(\frac{15}{I}-3J_h-\frac{15J_h}{I}+4J_h^2\right) \tag{4.66}$$

For complete derivation of T_h refer to Appendix (A).

When I is large, the term $\frac{12\rho_h}{I^2J_h^3}$ becomes insignificant because $I^2J_h^3$ is very large compared to $12\rho_h$. The same concept applied to term υ_h, hence:

$$T_h = KT\left(\frac{15}{I} - 3J_h - \frac{15J_h}{I} + 4J_h^2\right) \tag{4.67}$$

Assume cluster A and cluster B are two different clusters that belong to the same emitter. Assume the values of J_h for the the two clusters are respectively $J_A = a$, and $J_B = b$. Let us find the ratio of estimated parameters T_h of both clusters as follows:

$$R = \frac{KT\left(\frac{15}{I} - 3a - \frac{15a}{I} + 4a^2\right)}{KT\left(\frac{15}{I} - 3b - \frac{15J_h}{I} + 4b^2\right)} \tag{4.68}$$

$$R = \frac{15 - 15a - 3I\,a + 4I\,a^2}{15 - 15b - 3I\,b + 4I\,b^2} \tag{4.69}$$

For a realistic emitter scenario, let us assume that, for two different clusters of an emitter, the maximum value for the ratio $\frac{J_a}{J_b}$ is 4.

Now, the values of R_{dB} (i.e. $10\,log\,(R)$) will be calculated as shown in Table (4.1). The table is calculated such that J_a, and J_b can have values up to $J_a = 4$, and $J_b = 4$. This maximum value of J_a, and J_b was selected because there is good separation between the resultant R_{dB} values over this selected range of J_a and J_b (i.e. from 1 to 4). This good separation allows table (4.1) to be used as a lookup table. The value of I that was used in calculation is 200. The values in the table are almost fixed for a given range of I. For instance, when the value of I drops down to 50, then the values in table (4.1) change by only very small fraction (maximum of 1%).

	J_a			
J_b	1	2	3	4
1	0	-9.96731	-14.2894	-17.1412
2	9.96731	0	-4.32214	-7.1739
3	14.2894	4.32214	0	-2.85176
4	17.1412	7.1739	2.85176	0

Table 4.1: The value of R_{dB} for any two related clusters.

Now, for any two different clusters A, and B; if the value of R_{dB} is found to be matching one of the values in the table above, then A and B are related clusters. The term "related" means belonging to the same emitter. The value of R_{dB} provides information about the number of pulse positions that each cluster contributes within the emitter pattern. For example, take the pattern;

$$P = \{F1, F1, F1, F1, F2, F3, F3, F4, F4, F4, F4\}$$

For this pattern, $J_1 = 4$, $J_2 = 1$, $J_3 = 2$, and $J_4 = 4$ (these values and the pattern are unknown parameters). If the clusters that result from F1 and F2 respectively were examined, then $R_{dB} =$

7.1739. Looking up the value R_{dB} from the table, then J_1, and J_2 are obtained. If the value $R_{dB} = 0$ (as for the case of clusters F1-F4), then both clusters are related, and the value J_h of both clusters is identical, but no information about the value of J_h. Hence, the number of elements in the pattern P can be found if one cluster is compared with the rest of clusters, and if all found values of R_{dB} are non zero.

This is a new feature obtained from the TOA parameter. This feature can be used to resolve RF agility. The new feature can give information about the relationships between the different clusters. In fact, the above analysis has not considered the noise of TOA. As mentioned earlier, the values in the table change by a margin of a very small fraction. However, when noise is considered, then the values in the table will have a bigger margin, which leads to errors in the results, except for specific cases. In addition, by using the lookup table shown in table (4.1), the clusters that have J_a, or J_b that are greater than four will not be regarded as related clusters even if they are related to the same emitter, which add a limitation on using this feature in resolving RF agility.

Staggered PRI Emitter It does not make sense for an emitter to be staggered and frequency agile at the same time, except if this is used as some sort of a counter measure technique. Assuming such a counter measure could exist, then let us assume each pulse position in the staggered frame has a different frequency. If so, then the sequence of the emitter is a combination of different sequences. Each of these sequences has a PRI value that is equal to the frame-period. In other words, the clusters that result from this case have the same PRI value. Therefore, the agility of an emitter can be resolved based on the PRI analysis of clusters when the emitter combines RF agility and staggered PRI.

4.4.5.2 Utilizing TOA Difference

Each cluster is expected to have a few values for ∇TOA, if not only one value. For each cluster, the time difference ∇TOA between each pulse and the next pulse is calculated. A counter for each resultant ∇TOA value is established. This counter helps in confirming the value of ∇TOA. The confirmed ∇TOA values represent PRIs of the cluster T_C . PRI values of a cluster are given by;

$$T_C = N\,T_E \tag{4.70}$$

While, T_E is the PRI of the emitter, N is integer number such that $N \geq 1$. Assume T_{C1} and T_{C2} are the PRI values of two related clusters. Hence:

$$T_{C1} = N\,T_E \tag{4.71}$$

$$T_{C2} = M\,T_E \tag{4.72}$$

$$R_t = \frac{T_{C1}}{T_{C2}} = \frac{N\,T_E}{M\,T_E} = \frac{N}{M} \tag{4.73}$$

Two clusters are related only if the following condition is satisfied for any value of k:

$$e^{i(2\pi k\,R_t)} = 1 \tag{4.74}$$

Where k is an integer number such that, $k \geq 1$. Let us assume that, the maximum expected value for N and M is 10. Therefore, let us redefine k as $1 \leq k \leq 10$.

For each of the two clusters under test, an arbitrary confirmed PRI value is selected. The average value for each PRI is found in order to minimize the error in calculated R. Then $\cos(2\pi k\,R_t)$ is calculated for values of k from 1 to 10. If any value was found $\geq (1-\varepsilon)$, then clusters are related (ε is predetermined error margin).

In fact, this method is better than RMS method regarding its immunity to TOA noise, and to its lower complexity.

4.4.6 PDW Filter

After the successful de-interleaving of an emitter, the parametric boundaries of that emitter are sent to the PDW-Filter. This filter is used in order to block PDWs of de-interleaved emitters from the input stream of the de-interleaver. By blocking these PDWs, the unnecessary processing that results from repeatedly de-interleaving the same emitter is avoided.

In addition, emitters that have a continuous-pulse-stream add more load to the processing. Once these emitters are detected (by successful clustering), then their PDWs will be removed from the input buffer. This will significantly reduce the processing load on the de-interleaver, especially when these emitters have high PRF.

4.4.7 SNR Filter

As presented in an early chapter (see Chapter 2), the error margin of measured parameters under low SNR can be very large. It is desirable to keep the SNR above a certain threshold in order to

avoid increasing the error margin.

Assuming the channelized receiver has channel bandwidth B, then thermal noise power can be found by:

$$P_N = K_N T_N B \tag{4.75}$$

while, K_N and T_N are Boltzmann constant and absolute temperature, respectively.

Signal power can be found by:

$$P_{SN} = \frac{\sum_{n=0}^{N-1} E^2(n)}{N} \tag{4.76}$$

where E is the envelope of the time domain signal, N is the number of samples, and P_{SN} is signal and noise power combined. It is possible to implement this power calculation within EW receiver itself. Moreover, the power can be estimated from the amplitude parameter (A) of PDW. The amplitude parameter is the average value of the envelope of received pulse. Let us assume that the change of envelope value within the pulse is small. In this case, the power can be estimated by:

$$P_{SN} = A^2 \tag{4.77}$$

Assuming the noise power of the channel is the same as the thermal noise power of the channel, then:

$$P_s = A^2 - K_N T_N B \tag{4.78}$$

Hence, SNR can be given by:

$$SNR = \frac{A^2 - K_N T_B B}{K_N T_N B} \tag{4.79}$$

$$SNR = \frac{A^2}{K_N T_N B} - 1 \tag{4.80}$$

If the desired SNR value is known, then the minimum threshold for the amplitude can be calculated as follows:

$$A_{Th} = \sqrt{SNR\, K_N T_N B + 1} \tag{4.81}$$

This threshold can be set within the EW receiver, or alternatively it can be applied on the

amplitude parameter of PDW.

4.5 Conclusion

This chapter presented the proposed de-interleaving algorithm, which is able to handle the agility in time, as well as the agility in RF and PW. The proposed algorithm uses both a clustering stage and a time stage. DBSCAN is used in clustering stage because it is unsupervised, does not require prior information about clusters, is immune to noise, and has low complexity. The time stage uses time parameters in order to test clusters of potential agility relationships. The time of clusters is utilized for this purpose.

The proposed de-interleaving system is composed from different building blocks, with different functions. The de-interleaving solution provides detailed information about each de-interleaved emitter.

In this chapter, the test results of the proposed de-interleaving algorithm were discussed. In Chapter 6, the findings and conclusions of the dissertation will be provided.

Chapter 5

Results

This chapter present the test results of the proposed de-interleaving algorithm. The chapter consists of three sections. The results in the first section were used to support the decision to use DBSCAN in the clustering stage of the de-interleaving algorithm, in addition to supporting some other design decisions.

In the second section of this chapter, the proposed de-interleaving algorithm is tested against three test scenarios.

In the third section of this chapter, test samples for the proposed de-interleaving algorithm are given. The samples show a presentation of the interleaved PDWs. The results were compared with the simulated radar library, and the comparison results are visualized. The samples also present the Emitters-Tracks table, which is generated by the de-interleaving algorithm.

5.1 Clustering

The testing of the DBSCAN is discussed in this section. The test mainly focuses on the ability of the clustering algorithm to cluster radar pulses in the presence of outliers and measurement errors of the EW receiver. The test shows the accuracy of clustering using DBSCAN under these conditions and under different environment density scenarios.

5.1.1 Correct-Rate vs. Number of clusters

Clustering for interleaved pulses was performed using DBSCAN. The algorithm was tested against data that contained a number of clusters, up to 60 clusters. It is assumed that all clusters are formed under the condition of simultaneous reception from the various emitters (all emitters are

pointing towards the EW receiver at the same time). The parameters used are AOA and RF. The AOA centers of the clusters are uniformly distributed, and so are the RF centers. Each cluster has a Gaussian error distribution. The RMS error of the measured RF parameters and the AOA parameters are 1 MHz, and 5^o respectively. Each cluster, moreover has a random number of pulses which are uniformly distributed between 5 and 50. The simulation was repeated 1000 times for each of the test scenarios (where each scenario has different number of clusters). The simulation results are shown in Figure (5.1). From this figure, it is very clear that the correct clustering rate is very high. For example, the correct clustering rate with regard to 30 clusters is 99%. This corresponds to an average correct clustering of 29.7 clusters out of 30. In other words, the probability of correctly identifying all of the 30 clusters is very close to 100%.

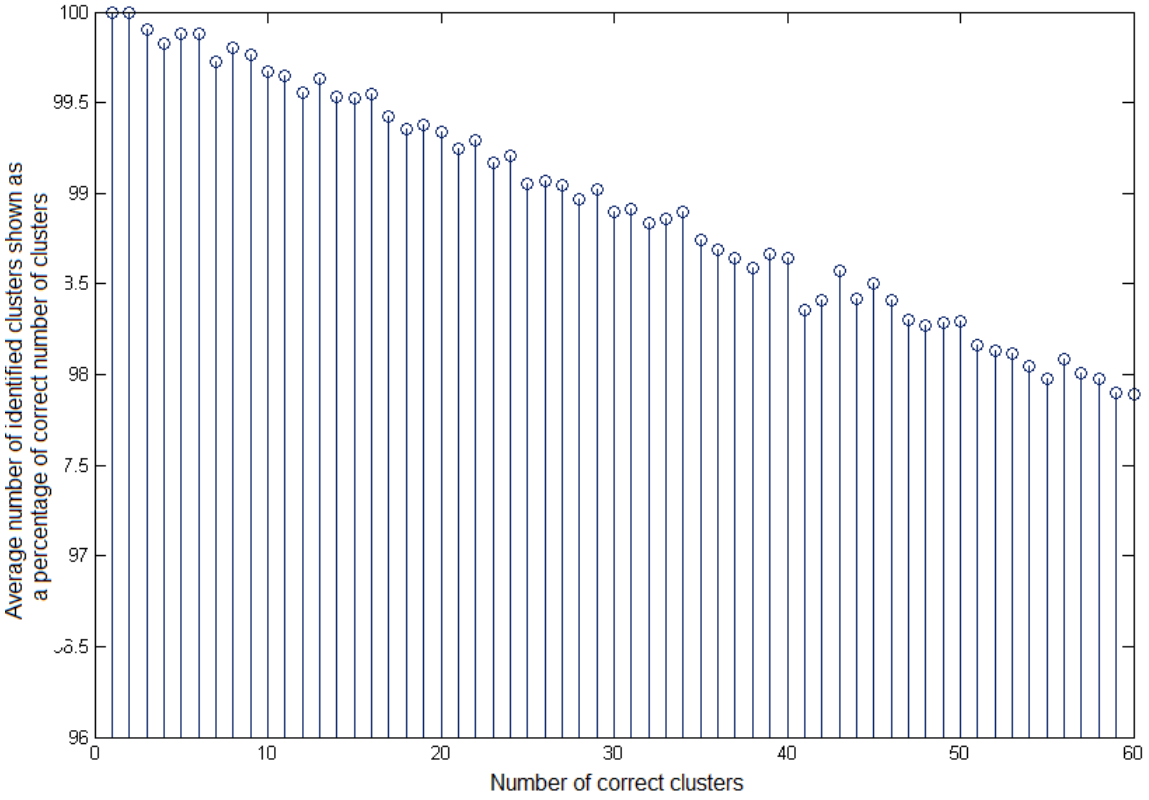


Figure 5.1: The average number of identified clusters (shown as a percentage of number of identified clusters) vs. Number of correct clusters. Note that y-axis is shown between 96 and 100.

Pulses that are not related to any of the real clusters are regarded as noise pulses or, specifically, outliers. No noise pulses exist in the simulation shown in Figure (5.1). The Simulation was repeated with 5% outlier pulses, and the result is shown in Figure (5.2). Taking 30 clusters, for example, the correct clustering rate is 98.88%. This result is very close to the previously shown result. In fact, the DBSCAN method is well known for its immunity to outliers, as was also confirmed here.

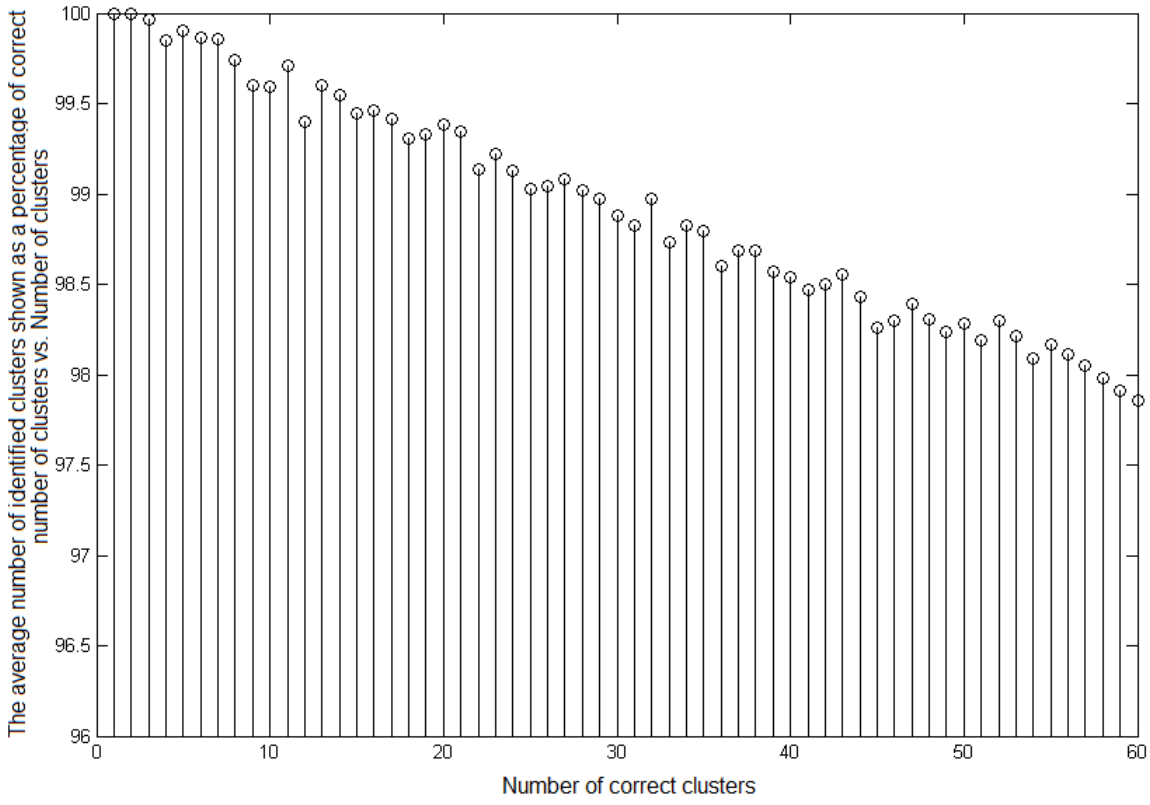


Figure 5.2: The average number of identified clusters (shown as a percentage of number of identified clusters) vs. Number of correct clusters. Note that y-axis is shown between 96 and 100. Data contain noise (outlier) pulses with ratio of 5 %.

To illustrate the performance of DBSCAN in clustering of the emitters' pulses, a random simulation was performed in an emitter environment that consisted of 30 clusters. All clusters were simultaneously received. The clustering results are shown in the Figure (5.3). Identified clusters are indicated by dots. The circles indicate the locations of the true pulses, while "x" marks indicate the outliers. Clusters are illustrated by means of various colours. The cluster ID is shown to the left of each identified cluster.

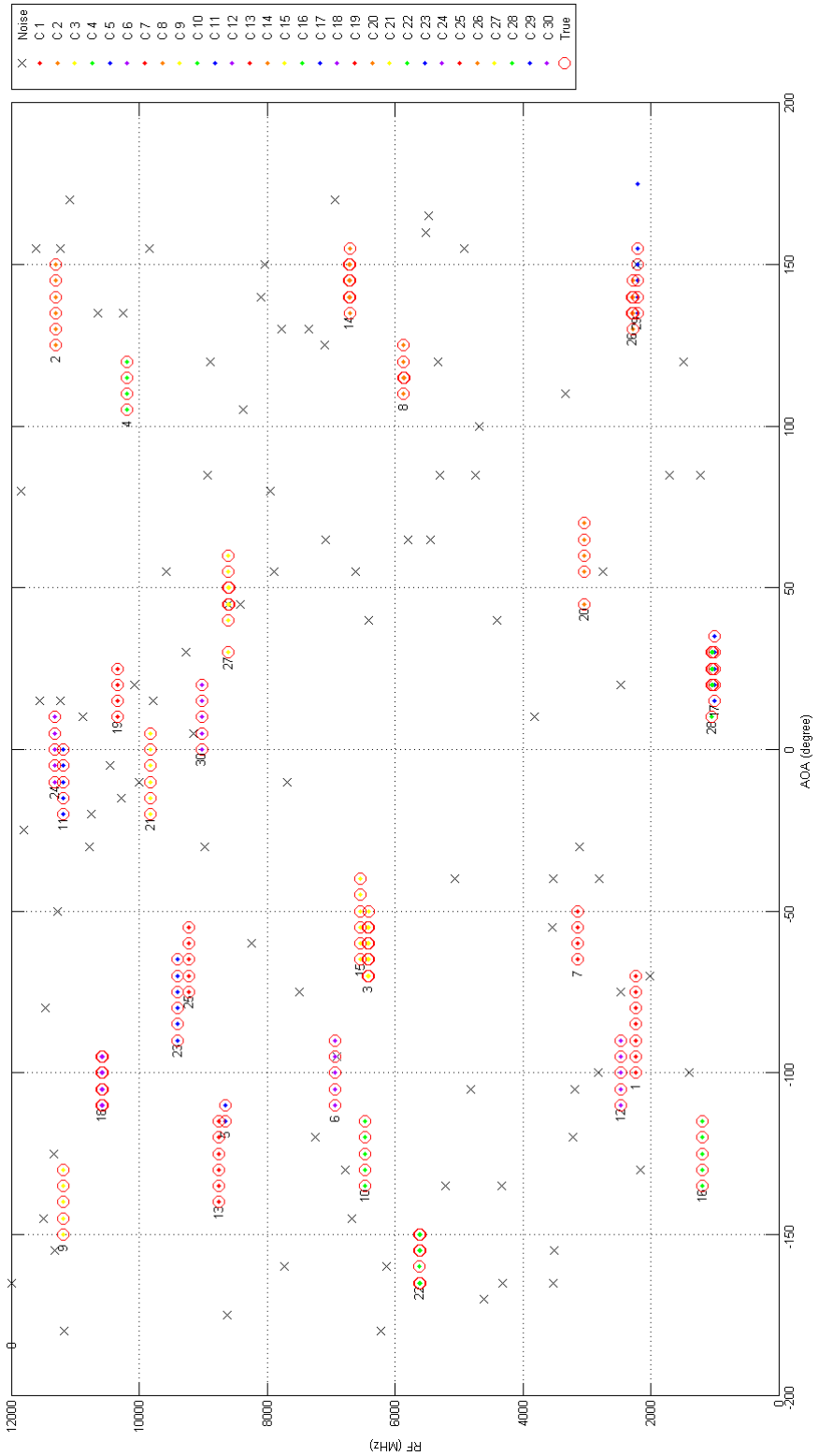
DBSCAN Clustering Conclusion In conclusion, DBSCAN showed a very good ability to cluster radar pulses with very high accuracy, in addition to its high immunity to noise. This clustering algorithm is thus very good for emitters that do not have agility in their frequency. However, for emitters that do have RF agility, DBSCAN will still be able to perform the correct clustering, but the algorithm cannot link the clusters that belong to the same emitter.

5.2 Algorithm Test

The algorithm was tested against three scenarios. The first scenario is for environments where all emitters are fixed frequency emitters. The second scenario is for environments where all emitters are frequency agile. The third scenario is for environments where some emitters are fixed, while others are agile. The measured RF parameters and AOA parameters of an emitter have a Gaussian distribution. The RMS error of the measured RF parameters and AOA parameters are 1 MHz, and 5^o respectively. The emitters are uniformly distributed in terms of their angle and frequency. The resolutions of the RF and AOA parameters are 1 MHz, and 5^o respectively. The following criteria were used in the test.

Exact-Number-of-Emitters If the de-interleaver has reported the exact number of emitters, then the condition Exact-Number-of-Emitters is met. The Exact-Number-of-Emitters-Rate is the rate for how many times this condition was met among 1000 different independent test scenarios. If the de-interleaver has reported either more or less emitters, then this condition is not met.

All-Emitters-Are-Reported This condition is met when at least all emitters in the environment (for a given test scenario) are reported by the de-interleaver. All-Emitters-Are-Reported-Rate is the rate for how many times this condition was met among 1000 different independent test scenarios.


Noise
C 1
C 2
C 3
C 4
C 5
C 6
C 7
C 8
C 9
C 10
C 11
C 12
C 13
C 14
C 15
C 16
C 17
C 18
C 19
C 20
C 21
C 22
C 23
C 24
C 25
C 26
C 27
C 28
C 29
C 30
True
RF (MHz)
AOA (degree)
12000
10000
8000
6000
4000
2000
0
200
150
100
50
0
-50
-100
-150
-200

Emitters Identification Rate This is the rate of correctly reported emitters during the 1000 different tests. This accumulates the number of correctly reported emitters during the 1000 different tests. In addition, it finds the total number of emitters during the 1000 different tests. Then, it uses these two values in the calculation.

To illustrate the difference between Emitters Identification Rate and Exact-Number-of-Emitters-Rate, let us give an example. The example is not realistic and only intended for illustration purposes. Let us assume that each test consists of 30 different emitters, and that there are 1000 different tests. Let us assume further that each test correctly reported 29 emitters out of 30 emitters. In this case, the condition Exact-Number-of-Emitters was never met. Therefore, the Exact-Number-of-Emitters-Ratio is zero ($^{0}/_{1000}$). The number of all emitters in the 1000 different tests amounts to 30,000 emitters, and the number of correctly reported emitters is 29,000. Therefore, the Emitters Identification Rate is $^{29,000}/_{30,000}$.

In the previous example, let us consider the scenario where we have 30 emitters and where each of the 1000 different tests reported 31 emitters instead of 30 emitters; while 30 emitters were correctly reported, and one emitter was incorrectly reported. In this case, the condition Exact-Number-of-Emitters was never met, but the condition All-Emitters-Are-Reported was met for all of the tests. Hence, the Exact-Number-of-Emitters-Ratio is zero, while the All-Emitters-Are-Reported-Ratio is $^{1000}/_{1000}$.

Average Speed In the previous chapter, it was explained that the PDW filter is part of the proposed de-interleaving algorithm. This PDW filter will block the pulses of successfully de-interleaved emitters at the input of the de-interleaver. This will significantly increase the speed of the algorithm, especially if one or more of the blocked emitters have a high PRF value; moreover, it will increase the speed if an emitters has a continuous stream of PDWs. This enhancement in speed is a result of dropping unnecessary pulses from processing, and this enhancement is strongly scenario dependent. Therefore, in order to provide a more meaningful figure, the average speed shown in the test below does not include the effect of the PDW filter.

Two option settings were used, viz., setting (I) and setting (II). Setting (II) tests if all the clusters of an emitter were received a sufficient number of times before reporting the emitter. A threshold parameter was defined for this purpose as the number of detections of a certain cluster. The threshold parameter, in other words, is the number of times that a certain cluster was detected. The used threshold condition was a minimum of two clusters. If the threshold condition was met, then the confirmation parameter takes the value of 1; otherwise, it takes the value of zero. In other words, only emitters with a confirmation parameter equal to one (confirmed) would be reported in

setting (II). Setting (I) would report emitters even if the confirmation parameter was zero. Setting (I) is equivalent to setting the threshold condition to one.

All tests were performed using Matlab® software which was running on a Personal Computer (PC) under a 64-bit Windows-7-professional operating system. The Central Processing Unit (CPU) of the PC was Intel® Core(TM) i7-3540M running at 3.00 GHz clock. The Random Access Memory (RAM) of the PC was 8 GB.

5.2.1 All Fixed Frequency Test

This test was performed using different numbers of fixed emitters. The number of fixed emitters takes the values 4, 8, 12, 16, 20, 24, and 28 emitters. The test was performed 1000 times for each of these values. Each test has random parameters for the emitters (as described at the beginning of Section (5.2)).

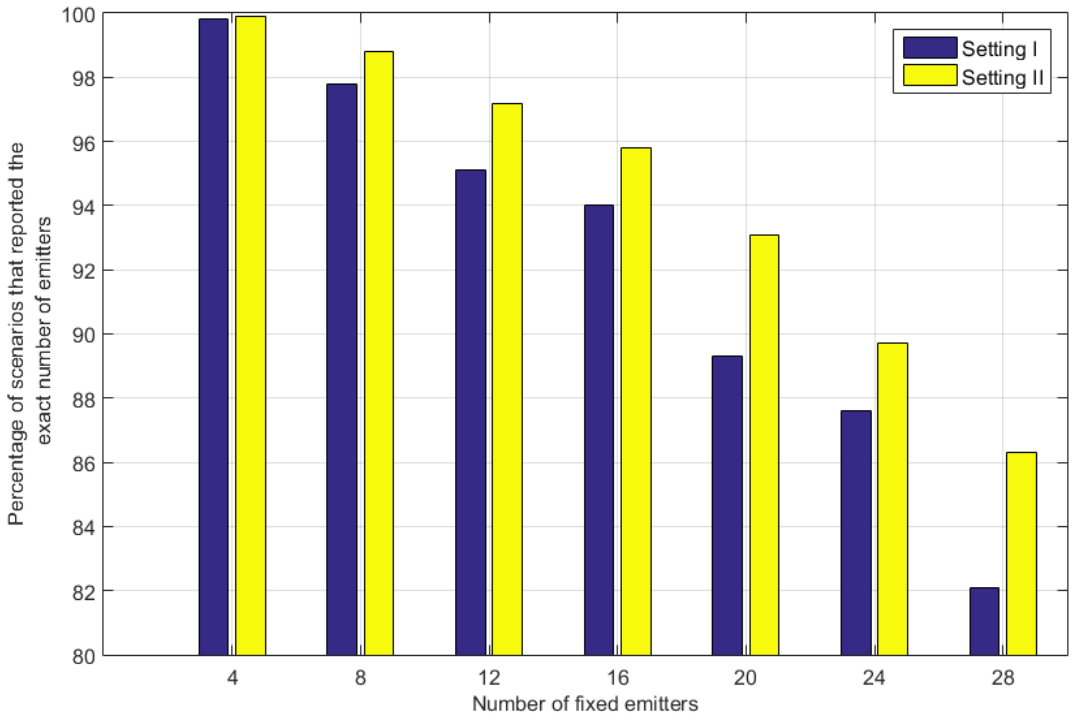


Figure 5.4: Percentage of tests that have reported the exact number of emitters (eaxact-number-of-emitters rate).

5.2.2 All Agile Frequency Test

This test was performed using different number of frequency agile emitters. The numbers of agile emitters takes the values 1, 2, 3, 4, 5, and 6. Each agile emitter consists of four different frequencies.

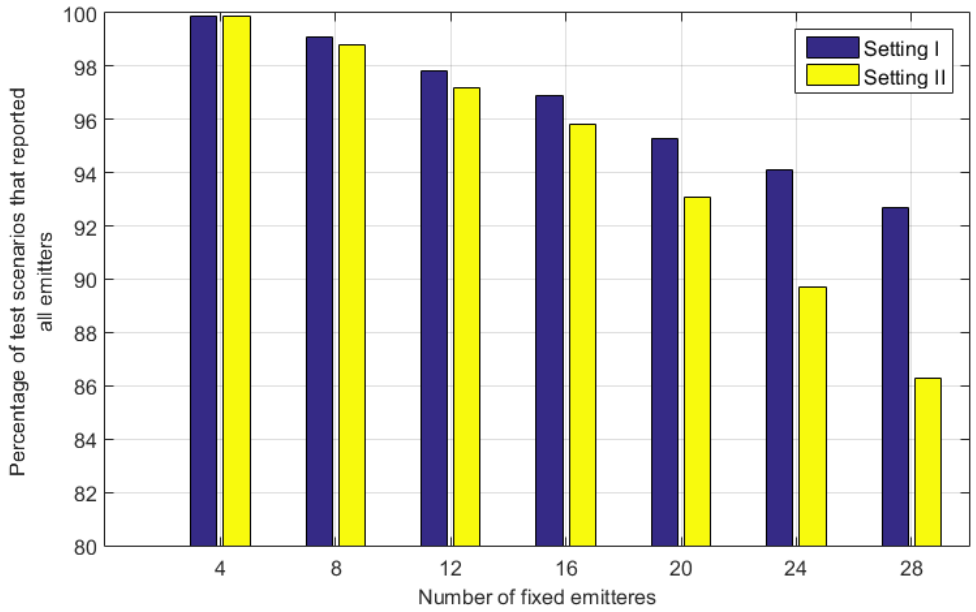


Figure 5.5: Percentage of tests that have reported all of the emitters (rate for the case where all emitters were correctly reported).

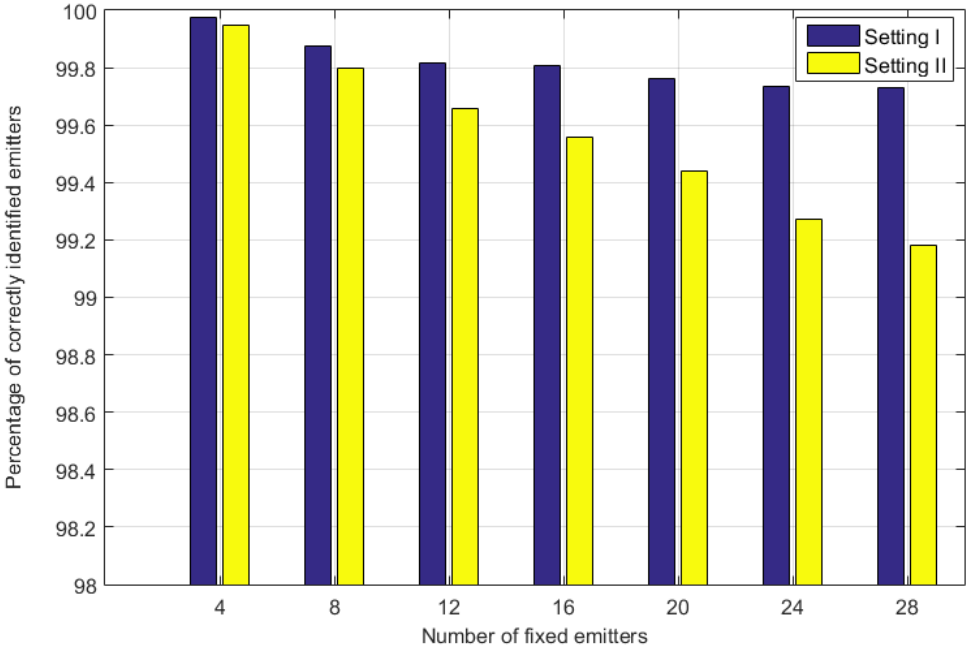


Figure 5.6: Percentage of correctly reported emitters (emitters' identification ratio).

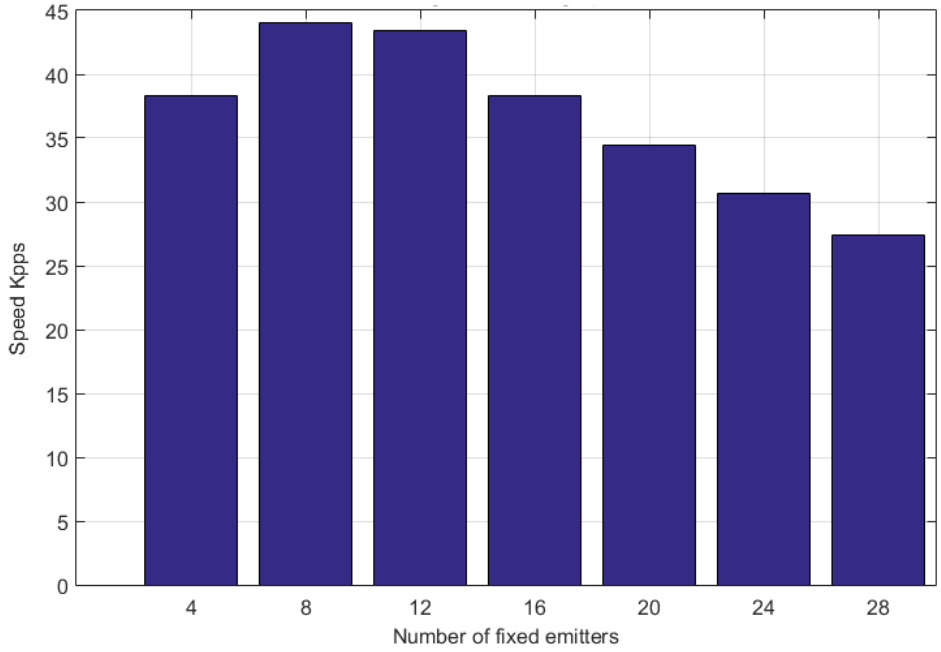


Figure 5.7: Average speed of the algorithm.

The test was performed 1000 times for each of these values. Each test has random parameters for the emitters (as described at the beginning of Section (5.2).

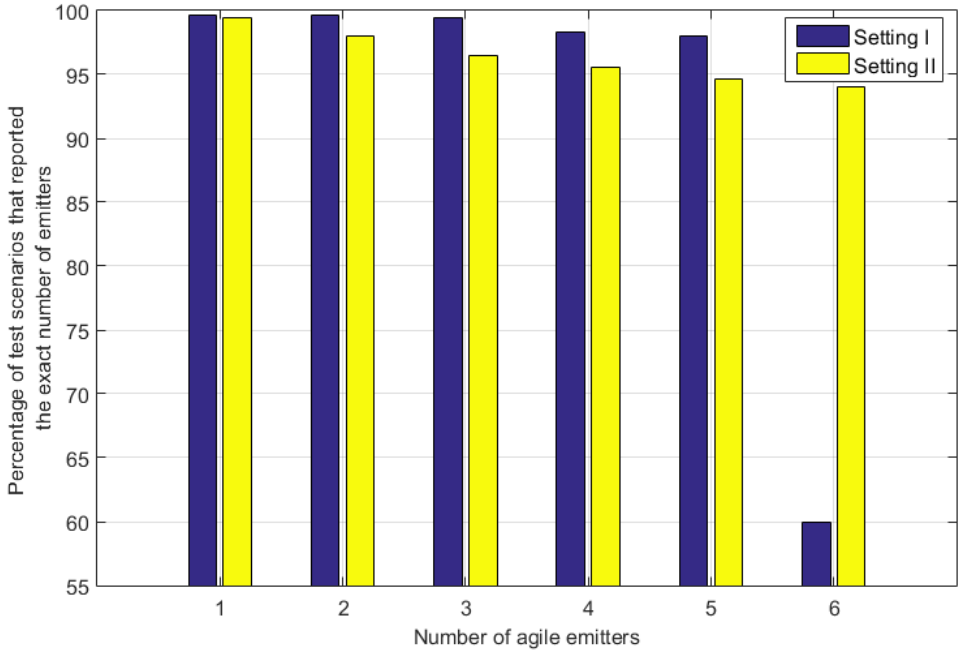


Figure 5.8: Percentage of tests that have reported the exact number of emitters (eaxact-number-of-emitters rate).

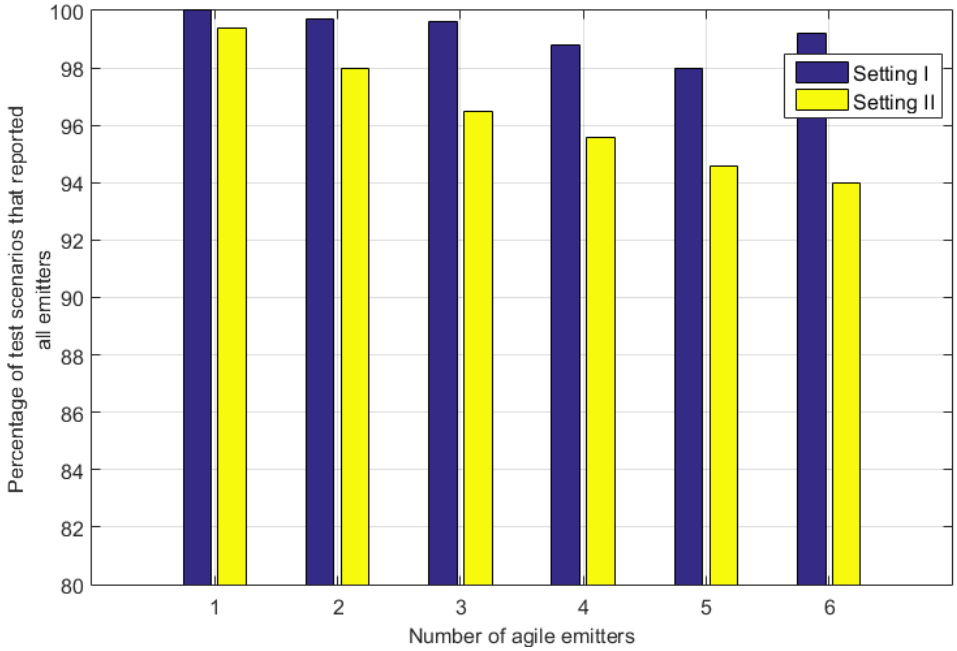


Figure 5.9: Percentage of tests that have reported all of emitters (rate for the case where all emitters were correctly reported).

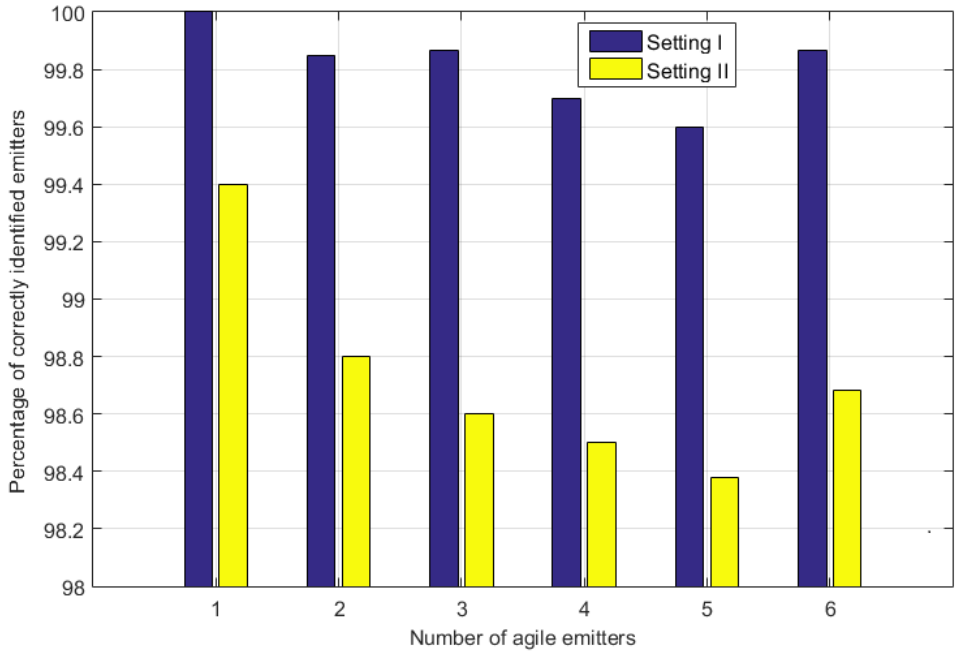


Figure 5.10: Percentage of correctly reported emitters (emitters' identification ratio).

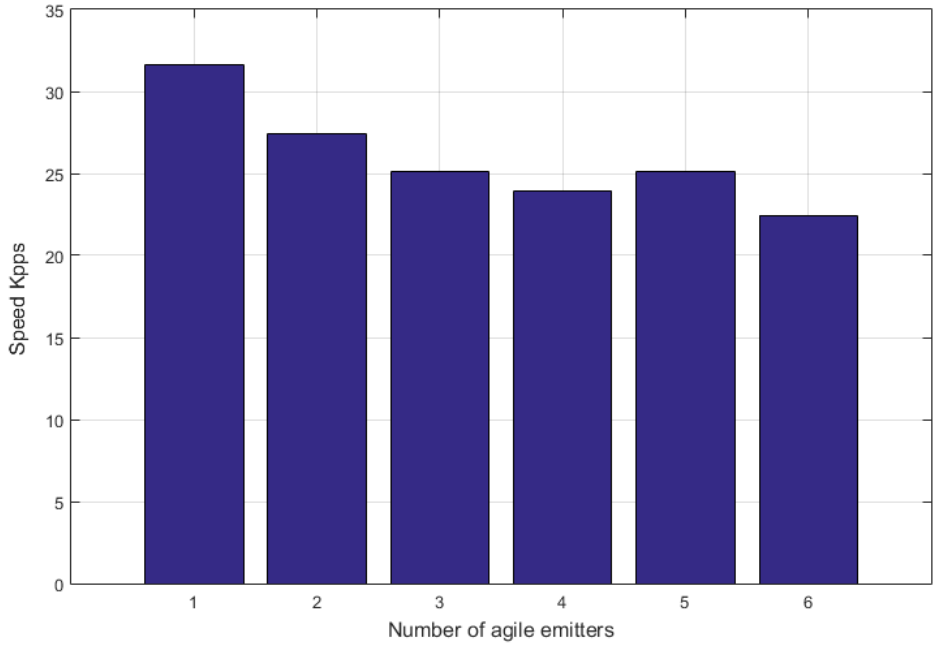


Figure 5.11: Average speed of the algorithm.

5.2.3 Combination of fixed and agile frequency test

This test was performed using combinations of fixed frequency emitters and frequency agile emitters. The numbers of fixed and agile emitters are shown in table (5.1). Each agile emitter consists of four different frequencies. The test was performed 1000 times for each of these values. Each test has random parameters for the emitters (as described at the beginning of Section (5.2). The centers of radio frequency of the emitters are generated using the uniform distribution as described in the beginning of Section (5.2). The four frequency components of each emitter are located within 10% around the center frequency. The four frequencies are equally spaced.

Test Setup ID	1	2	3	4	5	6	7
Fixed Emitters	1	10	20	1	3	5	6
Agile Emitters	1	1	1	4	4	4	4

Table 5.1: Test setup for combination of fixed and agile frequency emitters.

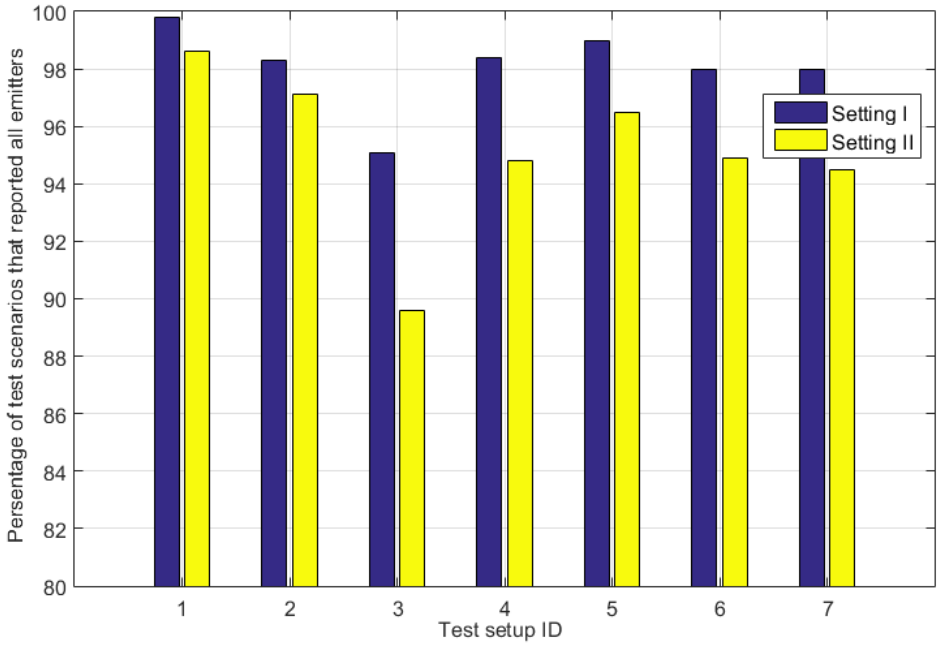


Figure 5.13: Percentage of tests that have reported all of emitters (rate for the case where all emitters were correctly reported).

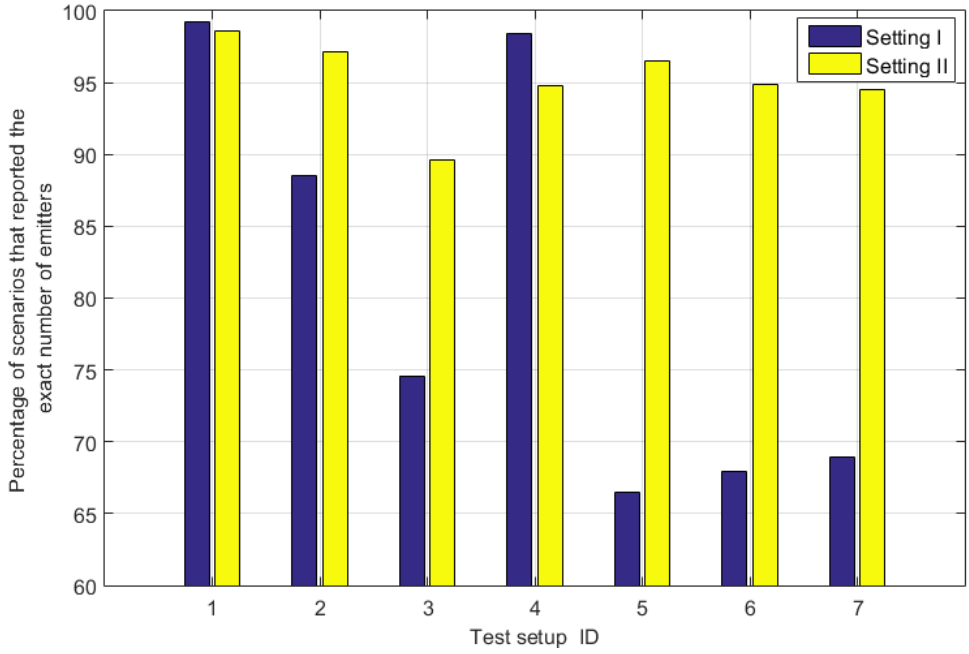


Figure 5.12: Percentage of tests that have reported the exact number of emitters (eaxact-number-of-emitters rate).

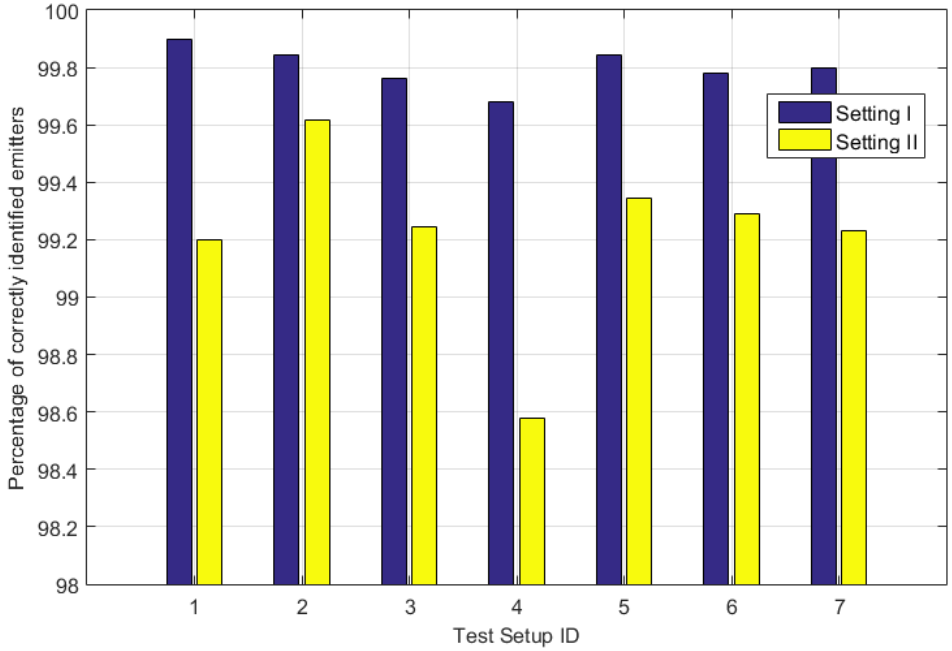


Figure 5.14: Percentage of correctly reported emitters (emitters' identification ratio).

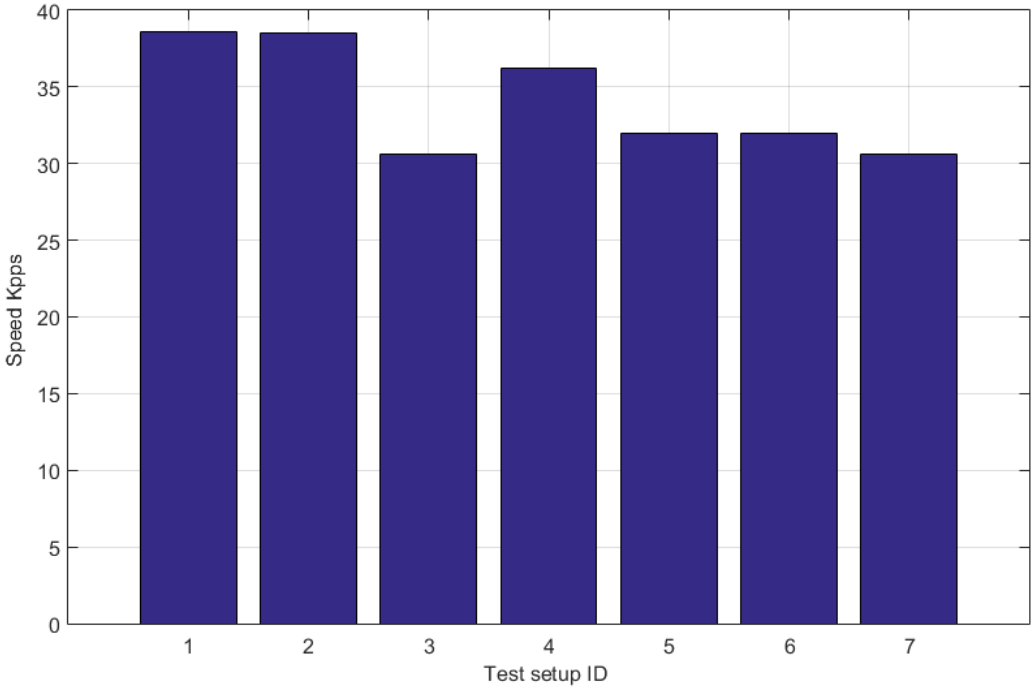


Figure 5.15: Average speed of the algorithm.

5.2.4 Discussion of Results

The first test criterion requires the algorithm to report all of the received emitters correctly, without reporting any false emitter. The test for fixed frequency emitters shows that the algorithm performs better in Setting (II) than Setting (I). Setting (II) only reports an emitter if it has at least two clusters formed at two different instances of time. This will reduce the chance of incorrectly generating additional clusters within the algorithm, which could happen due to the segmentation stage. Setting (I) requires at least one cluster for a given emitter in order to report that emitter. Therefore, the algorithm under Setting (I) is more susceptible to incorrectly generated clusters due to the segmentation stage.

The worst case for performance based on the first criterion was 82% using Setting (I) and 86% using Setting (II) for 28 clusters, in the case of the agile frequency emitter test. However, 99% accuracy is achievable for a lower number of emitters as shown in the case of four agile emitters.

The results discussed so far considered the cases for all fixed frequency emitters or all agile frequency emitters. When the emitters are a combination of fixed and agile emitters ,it was noticed that, as the number of fixed emitters increases in comparison to the number of agile emitters, then the performance of the algorithm can drop by a significant amount (from 98 % to around 67%) for the first criterion when Setting (I) is used. However, Setting (II) will maintain the performance of the algorithm with regard to the first criterion.

The reason behind the drop in the previous situation can be attributed to the agility resolving stage of the algorithm, which can mistakenly regard a fixed frequency emitter as part of an agile emitter. This could happen when the frequency of the fixed emitter is very close to the frequency of the agile emitter, and when they are, at the same time, close in angle. Reporting such an incorrect emitter will be regarded as a failure according to the first criterion.

The performance with regard to the second and third criteria is comparable to that of only fixed frequency emitters or only agile frequency emitters.

In fact, three different criteria were used for evaluating the algorithm in order to reflect the performance from different perspectives. Hence, a complete picture about the performance of the algorithm can be created by considering all the various perspectives. In addition, Setting (I) or Setting (II) can be chosen in order to optimize the performance of the algorithm according to which criteria is more important for the ELINT mission.

In addition to the three mentioned criteria, the achievable average speed of the algorithm was provided for each test. This average speed gives an indication of the number of pulses that can be handled in one second by the algorithm. It can further be seen from the results that, the speed

of the algorithm is higher when all emitters were fixed frequency emitters (an average of around 36 Kilo pulse per second (Kpps)). This average speed drops to 26 Kpps when all emitters were agile frequency emitters. The reason for this drop is attributed to the further processing that is performed by the algorithm when potential agile frequency emitters are encountered.

5.3 Test Samples

In this section, a sample of three different scenarios is presented. The first scenario is for three fixed frequency emitters. The interleaved PDWs are represented as shown in Figure (5.16). The radius in the polar graph represents the frequency parameter of the pulse, while the angle in the graph represents the AOA parameter of the pulse. The measured RF parameters and AOA parameters of an emitter have a Gaussian distribution. The RMS error of the measured RF parameters and AOA parameters are 1 MHz, and 5^o respectively. The resolutions of RF and AOA parameters are 1 MHz, and 5^o, respectively. After de-interleaving of the PDWs shown in Figure (5.16), using the proposed de-interleaving algorithm, the results are shown in Figure (5.17). Red circles in the figure represent the real emitters. The blue dots represent the detected emitters, based on de-interleaving. The number associated with the red circle is the ID assigned to the emitter in the simulation library. The number associated with the blue dot is the emitter track ID. This ID is assigned by de-interleaver to successfully de-interleaved emitters. Associating both ID numbers (i.e. Library ID, and Track ID) with each other is the task of classification. Simple classification is done here by comparing the parameters of Tracks with the parameters of the Emitters in the simulation library. As shown in the figure, the centers of the circles and the dots are not perfectly aligned. The reason for this is the resolution of the EW receiver, especially the AOA parameter, which has a resolution of 5^o. It is not the task of the de-interleaver to estimate the exact values of the parameters of the emitters, but the aim is to group related PDWs together.

The second scenario is for 30 fixed frequency emitters. Interleaved PDWs are shown in figure (5.18), while the de-interleaving results are shown in (5.19). The table of the Emitters Tracks generated by the de-interleaver is shown in Figure (5.20).

The third scenario is for 10 emitters, two of them are agile. Each agile emitter has 4 different frequencies. Interleaved PDWs are shown in Figure (5.21), while the de-interleaving results are shown in (5.22). The table of Emitters Tracks generated by de-interleaver is shown in Figure (5.23).

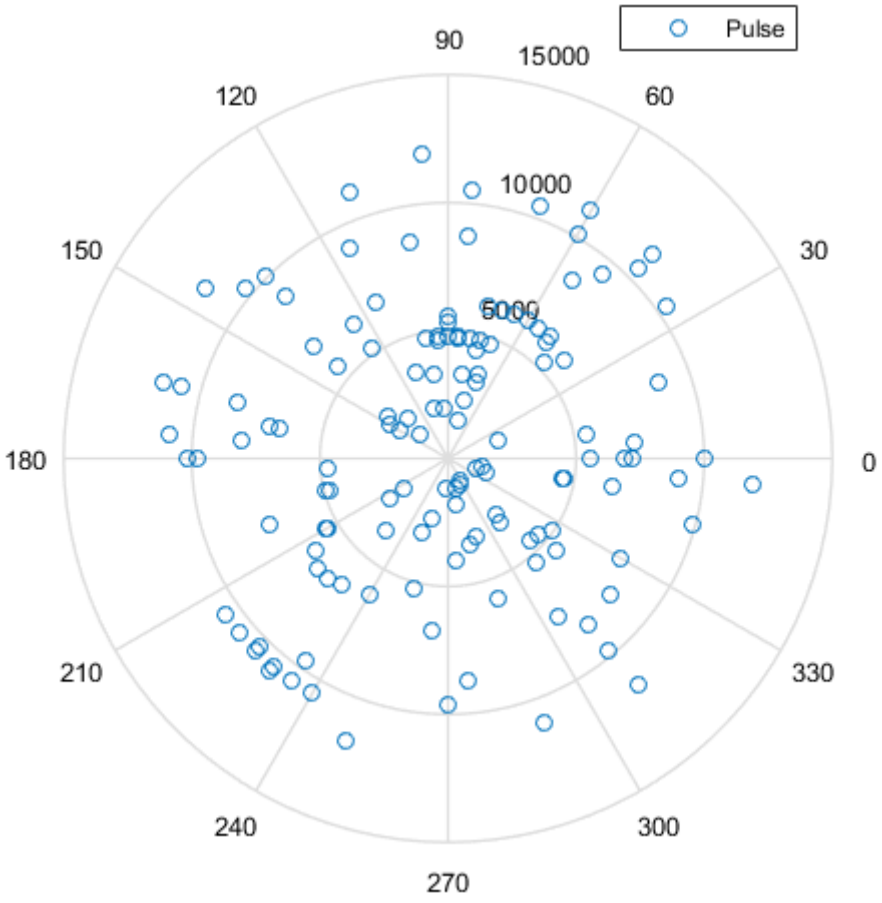


Figure 5.16: Interleaved PDWs input for three fixed frequency emitters. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

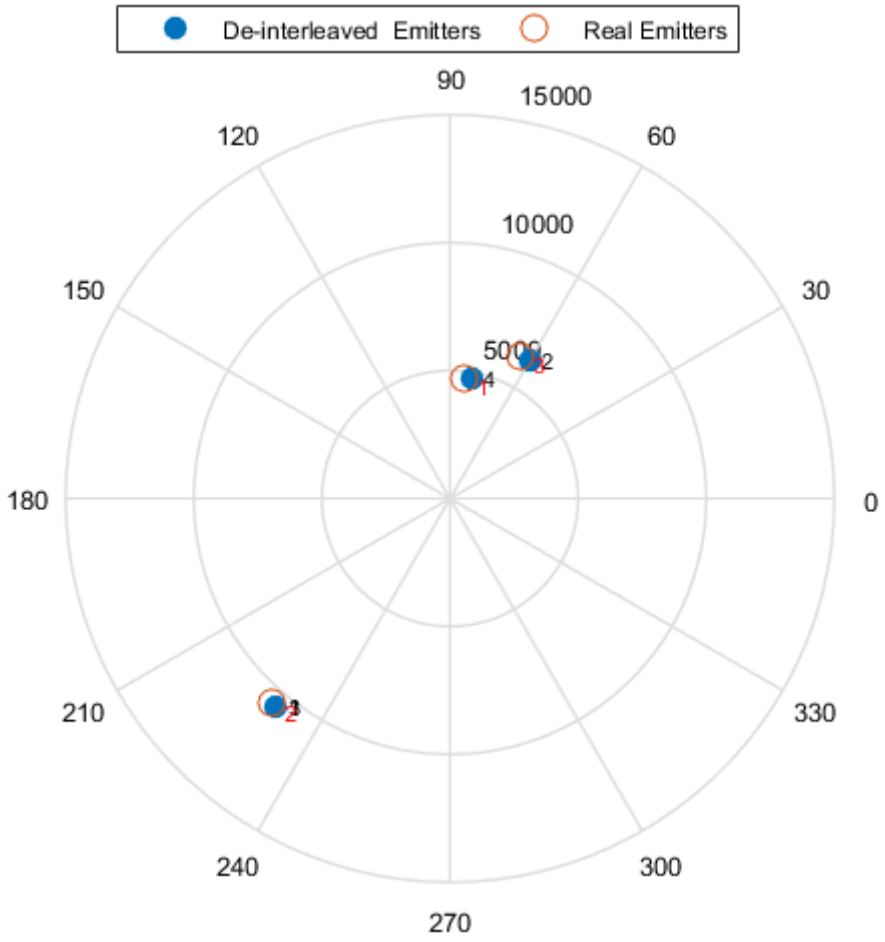


Figure 5.17: Results for de-interleaving of PDWs shown in Figure (5.16). The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

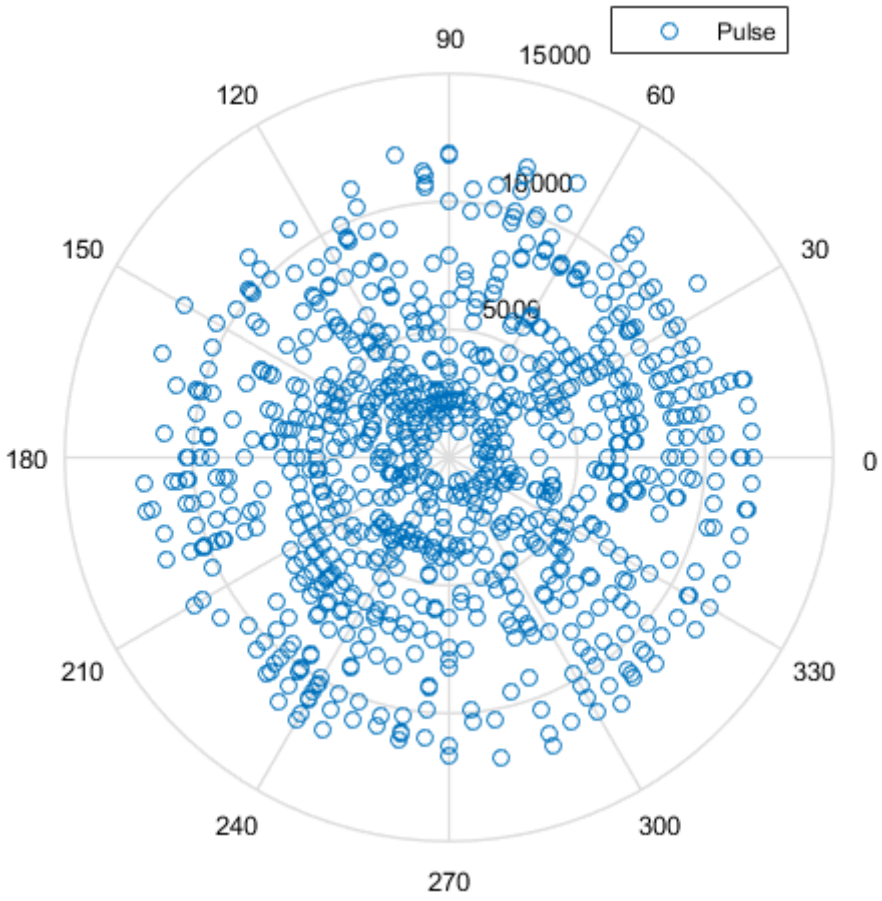


Figure 5.18: Interleaved PDWs input for 30 fixed frequency emitters. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

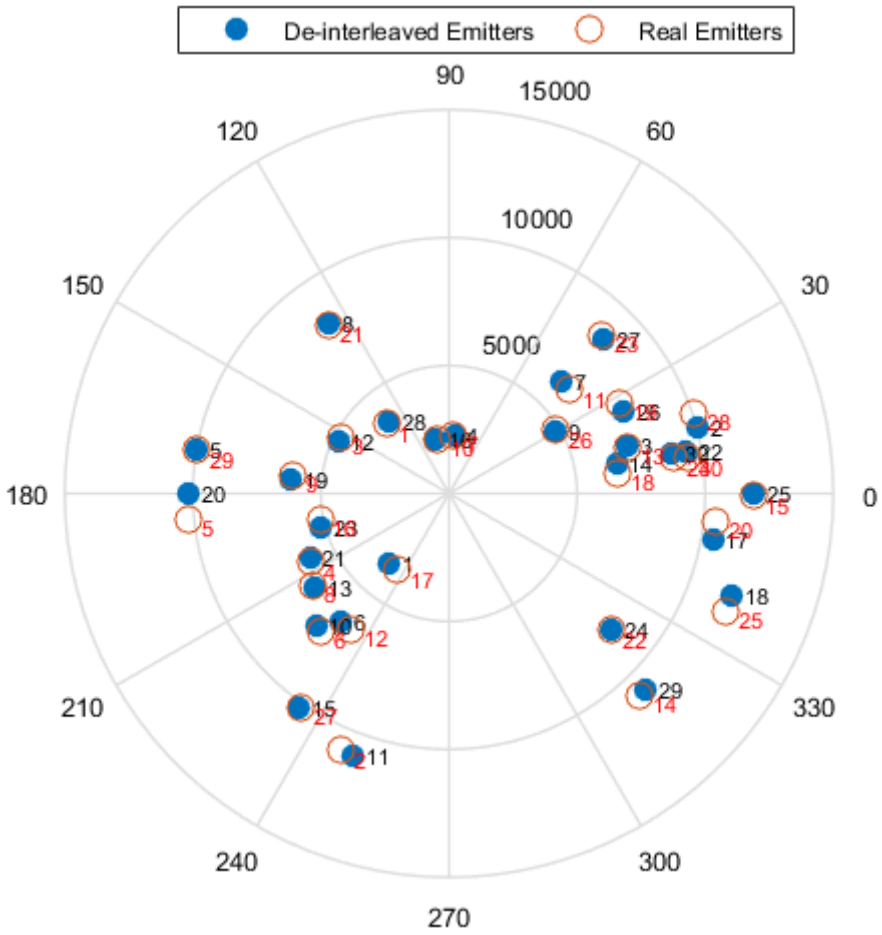


Figure 5.19: Results for de-interleaving of PDWs of second scenario. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

1x30 struct with 13 fields

Fields	ID	Library_ID	ClssesIDs	RF_center	RFagility	RFagility_steps	RF_hops	AOA_mean	Tracking	similarEmitters	similarity	clustersCounter	confirmed
1	1	17	1	3608	'Fixed'	[]	3608	-130	[]	[]	[]	10	1
2	2	28	2	10022	'Fixed'	[]	10022	15	[]	[]	[]	10	1
3	3	13	3	7220	'Fixed'	[]	7220	15	[]	[]	[]	10	1
4	4	7	4	2303	'Fixed'	[]	2303	85	[]	[]	[]	10	1
5	5	29	5	9980	'Fixed'	[]	9980	170	[]	[]	[]	10	1
6	6	12	6	6532	'Fixed'	[]	6532	-130	[]	[]	[]	10	1
7	7	11	7	6217	'Fixed'	[]	6217	45	[]	[]	[]	10	1
8	8	21	8	8124	'Fixed'	[]	8124	125	[]	[]	[]	10	1
9	9	26	9	4826	'Fixed'	[]	4826	30	[]	[]	[]	10	1
10	10	6	10	7347	'Fixed'	[]	7347	-135	[]	[]	[]	10	1
11	11	2	11	10886	'Fixed'	[]	10886	-110	[]	[]	[]	10	1
12	12	3	12	4790	'Fixed'	[]	4790	155	[]	[]	[]	10	1
13	13	8	13	6434	'Fixed'	[]	6434	-145	[]	[]	[]	10	1
14	14	18	14	6636	'Fixed'	[]	6636	10	[]	[]	[]	10	1
15	15	27	15	10191	'Fixed'	[]	10191	-125	[]	[]	[]	10	1
16	16	10	16	2179	'Fixed'	[]	2179	105	[]	[]	[]	10	1
17	17	20	17	10452	'Fixed'	[]	10452	-10	[]	[]	[]	10	1
18	18	25	18	11743	'Fixed'	[]	11743	-20	[]	[]	[]	10	1
19	19	9	19	6173	'Fixed'	[]	6173	175	[]	[]	[]	10	1
20	20	5	20	10198	'Fixed'	[]	10198	-180	[]	[]	[]	10	1
21	21	4	21	5995	'Fixed'	[]	5995	-155	[]	[]	[]	10	1
22	22	30	22	9378	'Fixed'	[]	9378	10	[]	[]	[]	10	1
23	23	16	23	5143	'Fixed'	[]	5143	-165	[]	[]	[]	10	1
24	24	22	24	8272	'Fixed'	[]	8272	-40	[]	[]	[]	10	1
25	25	15	25	11861	'Fixed'	[]	11861	0	[]	[]	[]	10	1
26	26	19	26	7511	'Fixed'	[]	7511	25	[]	[]	[]	10	1
27	27	23	27	8549	'Fixed'	[]	8549	45	[]	[]	[]	10	1
28	28	1	28	3668	'Fixed'	[]	3668	130	'Tracki...	[]	[]	1	1
29	29	14	29	10855	'Fixed'	[]	10855	-45	[]	[]	[]	10	1
30	30	24	30	8847	'Fixed'	[]	8847	10	[]	[]	[]	10	1

Figure 5.20: Emitter Tracks Table generated by de-interleaving algorithm. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

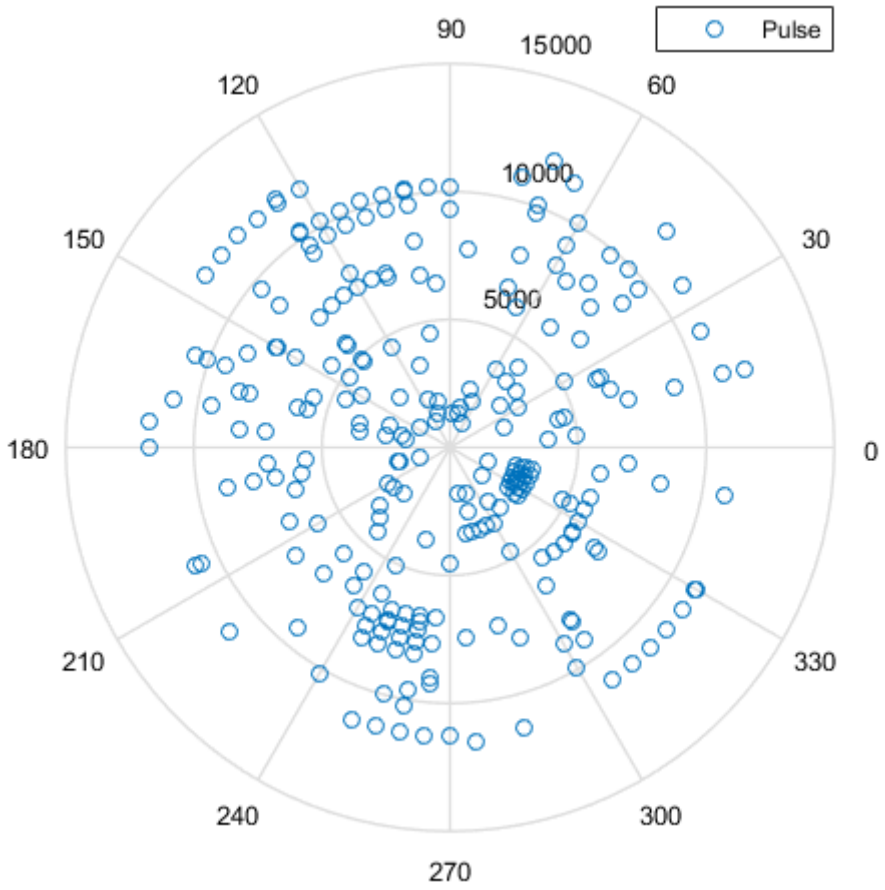


Figure 5.21: Interleaved PDWs input for 10 emitters, tow among them are frequency agile. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

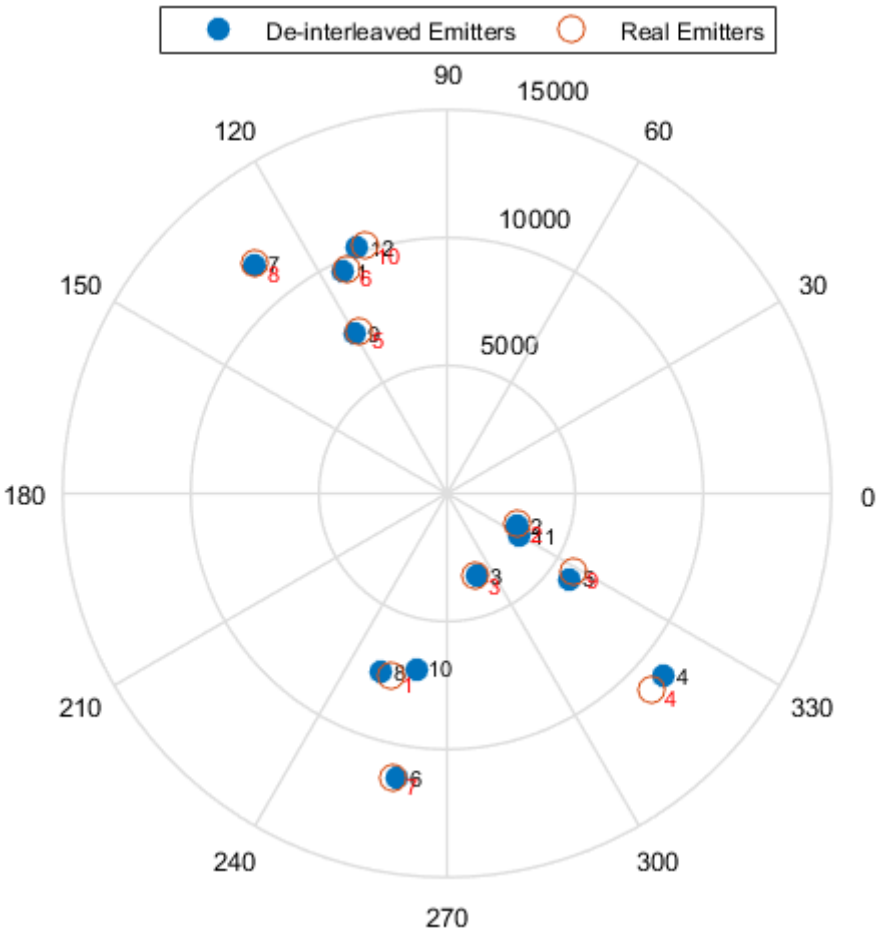


Figure 5.22: Results for de-interleaving of PDWs of third scenario. The angle and the radius axes of the polar plot are AOA and RF parameters, respectively.

1x10 struct with 13 fields

Fields	ID	Library_ID	clustersCount	ClssesIDs	RF_center	RFagility	RFagili	RF_hops	AOA_	Tracking	sim	sir	confirmed
1	1	6	5	1	9605	'Fixed'	[]	9605	115	[]	[]	[]	1
2	2	2	19	[2,3,4,5]	2990	'Agile'	4	[2691,2890,3091,3288]	-25	[]	[]	[]	1
3	3	3	5	6	3402	'Fixed'	[]	3402	-70	[]	[]	[]	1
4	4	4	5	7	11072	'Fixed'	[]	11072	-40	[]	[]	[]	1
5	5	9	5	8	5798	'Fixed'	[]	5798	-35	[]	[]	[]	1
6	6	7	5	9	11291	'Fixed'	[]	11291	-100	[]	[]	[]	1
7	7	8	5	10	11689	'Fixed'	[]	11689	130	[]	[]	[]	1
8	8	1	18	[11,12,13,14]	7449	'Agile'	4	[6703,7201,7697,8194]	-110	[]	[]	[]	1
9	9	5	5	15	7214	'Fixed'	[]	7214	120	[]	[]	[]	1
10	12	10	1	19	10206	'Fixed'	[]	10206	110	'Tracking'	[]	[]	1
11													

Figure 5.23: Emitter Tracks Table generated by de-interleaving algorithm.

5.4 Conclusion

The clustering stage of the algorithm shows good results in terms of the correct clustering rate. The correct clustering rate was very high (close to 99.90%) for a single cluster and the rate reduces as the number of clusters increases. However, the reduction in the correct clustering rate is small over the range from 1 to 60 emitters: the correct clustering rate was 98.88% for 30 clusters and close to 97.90% for 60 clusters.

The simulated test data included the measurement errors of the EW receiver and the outlier pulses. The RMS errors of the measured RF parameters and AOA parameters were 1 MHz and 5^o respectively, and the outlier ratio was 5%. The error has a Gaussian distribution. Each cluster moreover has a random number of pulses which are uniformly distributed between 5 and 50.

Based on the results, DBSCAN was found to be a very good choice for the clustering of radar signals, and it has a very high correct clustering rate. In order to achieve high performance results for the overall de-interleaving algorithm, it was decided to use less than 30 clusters, especially when the other stages of the algorithm are added.

In comparison to similar algorithms, the algorithm provided by Zhifu [35], for instance, used k-means clustering in its de-interleaving solution. The de-interleaving algorithm was tested against twelve fixed frequency emitters. The correct sorting rate was 99.64% according to [35]. The test result for equivalent number of clusters (12 clusters) for DBSCAN was 99.40%. However, the test performed for DBSCAN was extremely strict by assuming that, all of the emitters in the test were pointing toward the EW receiver in the same time. Better clustering results are achieved in the normal EW environment conditions, in other words, a 99.82% correct clustering rate, as can be seen in Figure (5.2).

DBSCAN was tested for 60 emitters (in contrast to only 12 emitters for k-means in [35]) with a very high correct clustering rate. The test of DBSCAN (provided in the dissertation) was performed by using 1000 independent run. In contrast, the results in [35] (which utilized k-means algorithm) appear to be based on single test. This is important to be mentioned because the algorithm provided in [35] can be very dependent on the test data. Moreover, the test performed for DBSCAN was extremely strict in that it assumed that all of the emitters in the test were pointing toward the EW receiver at the same time. Better clustering results can be achieved in the normal EW environment conditions as can be seen in Figure (5.6). The algorithm proposed in this dissertation has a correct clustering rate of more than 99.8% for 12 fixed frequency emitters compared to the correct clustering rate of 99.64% in Zhifu's algorithm.

The proposed de-interleaving in this dissertation deals with agility in frequency, in contrast

with Zhifu's algorithm which does not provide a solution for agile frequency emitters. This can be seen clearly from the second stage of Zhifu's algorithm which uses the RF parameter with the PW parameter in the clustering. Therefore, agility in RF becomes problematic for Zhifu's algorithm.

In this dissertation, therefore, the proposed algorithm provided a solution for resolving frequency agile emitters. The results of de-interleaving up to 6 radars each of which has four frequencies are presented in Figure (5.10), while the rate of successfully de-interleaving emitters is better than 99.6 %.

The proposed algorithm avoided utilizing the PW parameter in de-interleaving, and therefore PW agility is not a problem for the proposed algorithm. Hence, it was not necessary to perform the test in conditions of PW agility.

In the same sense, the PRI (or ∇TOA) parameter was not utilized in de-interleaving, but instead the time of cluster was used. The time of cluster is not sensitive to the agility of the PRI parameter, and therefore the test was not performed under this condition. Furthermore, the clustering stage does not even utilize any kind of time parameters.

The test verified the effectiveness of using DBSCAN in the clustering of radar pulses for the purpose of ELINT for online application. The results moreover verified the effectiveness of the proposed solution in de-interleaving with high accuracy, in a noisy and dense EW environment, where a variety of test scenarios was applied. The solution is effective for both fixed and agile radars, and it has met the requirements for the defined online ELINT application.

In this chapter, the test results of the proposed de-interleaving algorithm were presented. In Chapter 6, the conclusion of the dissertation will be provided.

Chapter 6

Conclusion

This dissertation proposed a de-interleaving solution for on-line ELINT application, specifically for ELINT/ESM receivers. The solution takes into consideration realistic EW hardware and actual EW environments. In addition, the dissertation provided a literature survey of the de-interleaving algorithms that could be utilized for on-line ELINT applications.

The dissertation contributed to this field by utilizing the DBSCAN clustering algorithm, which has not been used before in the problem of de-interleaving; and it was proved in this dissertation that it is effective in handling the de-interleaving problem. The de-interleaving solution presented herein is capable of handling agility in time, as well as agility in frequency and PW. At the same time, it is a feasible solution that does not require complex integration with an EW receiver, nor it is necessary to interfere with the signal processing unit of the EW receiver. The dissertation also discussed the on-line ELINT application that has not received attention in the literature.

The dissertation suggested specifications for a practical and feasible system, taking into account real EW systems. The importance of this work lies in supporting future research, by providing researchers with a literature survey that facilitates the selection of design decisions, when different requirements are proposed. Moreover, it provides a brief, but relevant EW introduction that is useful for of on-line ELINT de-interleaving.

The proposed algorithm was found to be effective in de-interleaving both fixed and frequency agile emitters. At the same time, the algorithm is able to deal with time agility. Moreover, it is capable of working in a dense emitter environment. The accuracy and speed of the algorithm were proven to meet the specified requirements, with a window for further enhancement of speed. The de-interleaving algorithms available in the literature were found to be unsatisfactory with regard to meeting the requirements of the recommended de-interleaving system described in Chapter 1.

The solution does however have a limitation with regard to the number of emitters, in that, it should not handle more than 30 simultaneous emitters (or clusters) in order to provide the best performance. However, even in the case of more than 30 simultaneous emitters, the performance does improve after a short time of running the de-interleaving algorithm. The algorithm can thus work with more than 30 emitters, with some compromise in performance.

In Chapter 1, it was assumed that, all emitters are stationary. In the case of moving emitters, the algorithm should be functional under some circumstances, but it was not designed or tested for this purpose.

The future work of the dissertation includes the following points:

- The proposed de-interleaving algorithm assumes a stationary EW receiver platform. Future work should take the movement of the platform into consideration in order to extend the use of the algorithm to include moving EW receiver platforms.

- The proposed algorithm achieved an average processing speed of around 36 Kpps for fixed frequency emitters. The future work should consider optimizing the implementation of the algorithm in order to enhance the average speed.

- The future work should increase the number of emitters that can be processed by the algorithm without sacrificing the processing speed or the performance that were achieved in the dissertation. The algorithm was able to handle the data of 60 clusters with a correct clustering rate of about 98%. If all emitters are fixed frequency emitters, then the clustering will be able to handle an equivalent number of emitters. However, because agile frequency emitters are also considered in the algorithm, the algorithm performs more analysis in the agility resolving stages. This processing limits the speed of the algorithm. At the same time, as the number of clusters increases, the chance of making incorrect decisions about frequency agility increases. The proposed de-interleaving algorithm in this dissertation is intended to handle 30 fixed frequency emitters with high performance. Moreover, it is intended to process agile frequency emitters with an equivalent number of clusters. The proposed future work is to increase this number without compromising the performance or the speed of the algorithm.

Bibliography

[1] D. C. Schleher, *Electronic warfare in the information age*. Artech House, Inc., 1999.

[2] D. Adamy, *EW 102: a second course in electronic warfare*. Artech House, 2004.

[3] A. De Martino, *Introduction to modern EW systems*. Artech House, 2012.

[4] J. Yan, H. Jiang, and Z. Lu, "Design and Implementation on Sorting and Tracking Processor for Radar Signal Based on DSP," in *Software Engineering (WCSE), 2013 Fourth World Congress on*, pp. 145–149, IEEE, 2013.

[5] R. M. Clark, *Intelligence analysis: a target-centric approach*. CQ press, 2012.

[6] X. Guo-liang, W. Hong-xun, X. Zhongwei, and W. Chao, "A fast sorting method for Modulated and Jitter PRI Radar Signals," in *Transportation, Mechanical, and Electrical Engineering (TMEE), 2011 International Conference on*, pp. 2210–2213, 2011.

[7] C. Davies and P. Hollands, "Automatic processing for ESM," in *IEE Proceedings F (Communications, Radar and Signal Processing)*, vol. 129, pp. 164–171, IET, 1982.

[8] G. Noone and S. D. Howard, "Investigation of periodic time series using neural networks and adaptive error thresholds," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, pp. 1541–1545, IEEE, 1995.

[9] R. G. Wiley, *ELINT: The interception and analysis of radar signals*. Artech House on Demand, 2006.

[10] R. Schmidt, "On separating interleaved pulse trains," *Aerospace and Electronic Systems, IEEE Transactions on*, no. 1, pp. 162–166, 1974.

[11] H. Mardia, "New techniques for the deinterleaving of repetitive sequences," in *Radar and Signal Processing, IEE Proceedings F*, vol. 136, pp. 149–154, 1989.

[12] D. Milojević and B. Popović, "Improved algorithm for the deinterleaving of radar pulses," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, pp. 98–104, IET, 1992.

[13] K. I. Nishiguchi and M. Kobayashi, "Improved algorithm for estimating pulse repetition intervals," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 36, no. 2, pp. 407–421, 2000.

[14] A. Mahdavi and A. M. Pezeshk, "A fast enhanced algorithm of PRI transform," in *Parallel Computing in Electrical Engineering (PARELEC), 2011 6th International Symposium on*, pp. 179–184, IEEE, 2011.

[15] C. Ren, J. Cao, Y. Fu, and K. E. Barner, "Improved method for pulse sorting based on PRI transform," in *SPIE Defense+ Security*, pp. 90911T–90911T, International Society for Optics and Photonics, 2014.

[16] S. Wu, W. Su, L. Zhang, and X. Shan, "The Smoothing Method for Finding PRIs from the PRI Transform Spectrum," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, pp. 1–6, IEEE, 2008.

[17] K. Nishiguchi, "Time-period analysis for pulse train deinterleaving," *Trans. Computers of the Society of Instrument and Control Engineers*, vol. 4, pp. 68–78, 2005.

[18] J. Perkins and I. Coat, "Pulse train deinterleaving via the Hough transform," in *icassp*, pp. 197–200, IEEE, 1994.

[19] M. Jian and H. Laizhao, "A new transform method for pulse train parameter estimation," in *Signal Processing Proceedings, 1998. ICSP'98. 1998 Fourth International Conference on*, vol. 2, pp. 1562–1565, IEEE, 1998.

[20] Z. Guoyi, Z. Xuzhou, and C. Shuo, "A PRI jitter signal sorting method based on cumulative square sine wave interpolating," in *Mechatronic Sciences, Electric Engineering and Computer (MEC), Proceedings 2013 International Conference on*, pp. 921–924, IEEE, 2013.

[21] S. Dai, W. Lei, Y. Cheng, and D. Wang, "Clustering of DOA data in radar pulse based on SOFM and CDbw," *Journal of Electronics (China)*, vol. 31, no. 2, pp. 107–114, 2014.

[22] C. Zhao, Y. Zhao, and J. Lu, "Radar signals sorting with kohonen neural net," in *Signal Processing, 2006 8th International Conference on*, vol. 3, 2006.

[23] Z.-X. Yang, Y.-q. Zhu, X.-N. Li, and S. Wei, "Sorting radar emitter signal based on wpt6 and cr1," in *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*, pp. 1–4, IEEE, 2009.

[24] H. Li, B. Chen, J. Han, and W. Dong, "A new method for sorting radiating-source," in *2009 International Conference on Networks Security, Wireless Communications and Trusted Computing*, vol. 2, pp. 817–819, 2009.

[25] Z. Wang, D. Zhang, D. Bi, and S. Wang, "Multiple-parameter radar signal sorting using support vector clustering and similitude entropy index," *Circuits, Systems, and Signal Processing*, vol. 33, no. 6, pp. 1985–1996, 2014.

[26] Q. Guo, X. Zhang, and Z. Li, "A novel sorting method of radar signals based on support vector clustering and delaminating coupling," in *Cognitive Informatics, 2006. ICCI 2006. 5th IEEE International Conference on*, vol. 2, pp. 839–844, IEEE, 2006.

[27] Q. G. X. Z. Z. Li, "a novel joint de-interleaving/recognition system of radar pulse sequence," Innovative Computing, Information and Control, 2006. ICICIC'06. First International Conference, 2006.

[28] Q. Guo, W.-H. Chen, X.-Z. Zhang, and Z. Li, "Multiple-Parameter De-Interleaving System in ESM Data Processing Scheme," in *2005 International Conference on Machine Learning and Cybernetics*, pp. 2497–2502, 2006.

[29] Q. Guo, X. Zhang, and Z. Li, "Cascade coupling and support vector clustering based novel sorting method of radar pulses," in *Circuits and Systems, 2006. MWSCAS'06. 49th IEEE International Midwest Symposium on*, vol. 2, pp. 351–355, IEEE, 2006.

[30] Q. Guo, W. Xu, C. Wang, and D. Guan, "Segment clustering radar signal sorting," in *Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on*, vol. 1, pp. 943–946, IEEE, 2009.

[31] G. Noone, "Fuzzy art variants for improved single pass sequential clustering of data," 2005.

[32] A. Ata'a and S. Abdullah, "Deinterleaving of radar signals and prf identification algorithms," *Radar, Sonar and Navigation, IET*, vol. 1, no. 5, pp. 340–347, 2007.

[33] M. Thompson and J. C. Sciortino, "Implementation and testing of fuzzy adaptive resonance theory algorithm to analyze complex radar data sets," *Transactions of the AOC*, vol. 1, no. 1, 2004.

[34] M.-A. Cantin, Y. Blaquière, Y. Savaria, E. Granger, and P. Lavoie, "Implementation of the fuzzy ART neural network for fast clustering of radar pulses," in *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*, vol. 2, pp. 458–461, IEEE, 1998.

[35] Y. Zhifu, Y. Fei, and L. Jingqing, "A multi-parameter synthetic signal sorting algorithm based on clustering," in *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, pp. 2–363, IEEE, 2007.

[36] J. Ma, G. Huang, D. Zhou, B. Fang, and H. Yu, "Underdetermined blind sorting of radar signals based on sparse component analysis," in *Communication Technology (ICCT), 2012 IEEE 14th International Conference on*, pp. 1296–1300, IEEE, 2012.

[37] Z. Ming, W. Sidong, F. Kechang, and J. Weidong, "Radar signals clustering based on spectrum atom decomposition and kernel method," in *Measurement, Information and Control (ICMIC), 2013 International Conference on*, vol. 2, pp. 843–846, IEEE, 2013.

[38] J. Liu, J. P. Lee, L. Li, Z.-Q. Luo, and K. M. Wong, "Online clustering algorithms for radar emitter classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 8, pp. 1185–1196, 2005.

[39] Z. Yongqiang and S. Guozhi, "Application of radial basis function neural networks in complicated radar signal measurement and sorting," in *Electronic Measurement and Instruments, 2007. ICEMI'07. 8th International Conference on*, pp. 3–375, IEEE, 2007.

[40] C. Ting, L. Jingqing, *et al.*, "A radar signal sorting method based on immune evolutionary artificial neural network," in *Wireless Communications, Networking and Mobile Computing, 2008. WiCOM'08. 4th International Conference on*, pp. 1–4, IEEE, 2008.

[41] Y. Chan, F. Chan, and H. Hassan, "Performance evaluation of ESM deinterleaver using TOA analysis," in *Microwaves, Radar and Wireless Communications, 2002. MIKON-2002. 14th International Conference on*, vol. 2, pp. 341–350, IEEE, 2002.

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

[43] J. Tsui, *Digital techniques for wideband receivers*. SciTech Publishing, 2004.

[44] M. Stiglitz, "Active radar electronic countermeasures," *Microwave Journal*, vol. 33, no. 7, pp. 139–140, 1990.

[45] M. I. Skolnik, "Radar handbook," 1970.

[46] F. E. Nathanson, J. P. Reilly, and M. N. Cohen, "Radar design principles-signal processing and the environment," *NASA STI/Recon Technical Report A*, vol. 91, 1991.

[47] J. Tsui, *Microwave receivers with electronic warfare applications*. IET, 2005.

[48] J. B. Tsui, *Special design topics in digital wideband receivers*. Artech House, 2009.

[49] G. Noone and S. D. Howard, "Deinterleaving radar pulse trains using neural networks," *Salisbury, South Australia: Defence Science and Technology Organisation 71L PO Box*, vol. 1500, 1995.

[50] J. Cao, F. Li, F. Yang, and L. Ren, "Pulse deinterleaving using isomorphic subsequences," IET, 2015.

[51] Y. Kuang, Q. Shi, Q. Chen, L. Yun, and K. Long, "A simple way to deinterleave repetitive pulse sequences," in *7th WSEAS Int. Conf. on Mathematical Methods and Computational Techniques in Electrical Engineering, Sofia*, pp. 218–222, 2005.

[52] B. E. S. Bildy, "Satellite cluster consepts a system evaluation with emphasis on deinterleaving and emitter recognition," Master's thesis, Norwegian University of Science and Technology, 2013.

[53] G. Guo, Y. Mao, J. Han, and X. Qing, "An improved algorithm of PRI transform," in *Intelligent Systems, 2009. GCIS'09. WRI Global Congress on*, vol. 3, pp. 145–149, IEEE, 2009.

[54] P. Wei, H. Wu, and L. Qiu, "The sorting of radar pulse signal based on plane transformation," in *Systems and Informatics (ICSAI), 2012 International Conference on*, pp. 326–328, 2012.

[55] H. Mardia, "Adaptive multidimensional clustering for ESM," in *Signal Processing for ESM Systems, IEE Colloquium on*, pp. 5–1, IET, 1988.

[56] P. C. Wang, M. Orr, M. Sparrow, and M. Apa, "System and method for detecting and deinterleaving radar emitters," July 2008. US Patent 7,397,415.

[57] P. C. Wang, M. Orr, M. Sparrow, and M. Apa, "System and method for detecting and deinterleaving radar emitters using snapshots of varying lengths," Nov. 2010.

[58] J. A. Sirois, "Detection and identification of stable pri patterns using multiple parallel hypothesis correlation algorithms," Nov 2006. US 7,133,887 B2.

[59] J. A. Sirois, "Detection and identification of stable pri patterns using multiple parallel hypothesis correlation algorithms," feb 2005. US 2005/0033789 A1.

[60] D. P. Andersen and S. A. Murphy, "Radar pulse detection and classification system," Dec. 1996. US Patent 5,583,505.

[61] J. Pandu, N. Balaji, and C. Naidu, "FPGA implementation of multi parameter deinterleaving," in *Electronics and Communication Systems (ICECS), 2014 International Conference on*, pp. 1–5, IEEE, 2014.

[62] G.-r. Quan, Y.-S. Sun, and B. Chen, "Folding Deinterleaving Algorithm for Multiple Mixed Pulse Trains with Pulse Repetition Intervals," in *Machine Learning and Cybernetics, 2006 International Conference on*, pp. 3334–3338, IEEE, 2006.

[63] P. A. Valand, "Method, use of said method and arrangements in an electronic support measures system," sep 2011. US Patent 8,013,782.

[64] K. R. Jenkin, "Real-time pulse processor," Jan. 1988. US Patent 4,721,958.

[65] I. Zelinka, G. Chen, O. E. Rössler, V. Snasel, and A. Abraham, *Nostradamus 2013: Prediction, Modeling and Analysis of Complex Systems*, vol. 210. Springer Science and Business Media, 2013.

[66] T. Smith and D. Alahakoon, "Growing self-organizing map for online continuous clustering," in *Foundations of Computational Intelligence Volume 4*, pp. 49–83, Springer, 2009.

[67] N. Rougier and Y. Boniface, "Dynamic self-organising map," *Neurocomputing*, vol. 74, no. 11, pp. 1840–1847, 2011.

[68] Q. Guo, X. Zhang, and Z. Li, "SVC & k-means and type-entropy based deinterleaving/recognition system of radar pulses," in *Information Acquisition, 2006 IEEE International Conference on*, pp. 742–747, IEEE, 2006.

[69] M. Zhu, K. Fu, X. Huang, S. Wu, and W. Jin, "A novel recognition approach for radar emitter signals based on on-line independent support vector machines," *Advances in Computer Science and its Applications*, vol. 2, no. 3, pp. 390–396, 2013.

[70] W. Yi, "A novel method for sorting unknown radar emitter," *Microcomputer Information*, vol. 4, p. 086, 2010.

[71] J. Han, M.-h. He, Y.-q. Zhu, and B.-g. Zhu, "A novel method for sorting radar radiating-source signal based on ambiguity function," in *Networks Security, Wireless Communications and Trusted Computing, 2009. NSWCTC'09. International Conference on*, vol. 2, pp. 820–823, IEEE, 2009.

[72] J. Wang, Q. Shen, and K.-b. Qin, "A novel method for sorting radar emitter signal based on the bispectrum," in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, pp. 1–4, IEEE, 2009.

[73] G. Yu, J. Han, and Y. Mao, "A new method for sorting unknown radar emitter signals," in *Signal Processing, 2008. ICSP 2008. 9th International Conference on*, pp. 2346–2350, 2008.

[74] G. HAMMACK, "Method for de-interleaving received radar pulses using dynamically updated weights," Jan. 2014. WO Patent App. PCT/US2013/033,387.

[75] M. Keshavarzi and A. M. Pezeshk, "A simple geometrical approach for deinterleaving radar pulse trains," in *Computer Modelling and Simulation (UKSim), 2016 UKSim-AMSS 18th International Conference on*, pp. 172–177, IEEE, 2016.

[76] J. Liu, S. Gao, Z.-Q. Luo, T. Davidson, and J. P. Lee, "The minimum description length criterion applied to emitter number detection and pulse classification," in *Statistical Signal and Array Processing, 1998. Proceedings., Ninth IEEE SP Workshop on*, pp. 172–175, IEEE, 1998.

[77] M. P. Card and P. G. Springer, "Method, system and program product for deinterleaving and classifying arbitrary radar pulse patterns using non-determinsitic finite state automata," Nov. 2013. US Patent 8,587,468.

[78] W. T. Horn and S. F. Hurt, "Method and system for deinterleaving," Jan. 2006. US Patent 6,985,102.

[79] S. H. Arneson and M. A. Nayfeh, "Method and system for collecting information about a plurality of emitters," Nov. 2000. US Patent 6,147,646.

[80] C. D. Wang and J. Thompson, "Real time adaptive probabilistic neural network system and method for data sorting," Jan. 1994. US Patent 5,276,772.

[81] M. W. Maier, "Associative hierarchiacl deinterleaver," Apr. 1990. US Patent 4,918,455.

[82] M. W. Maier, "Associative hierarchiacl deinterleaver," apr 1990. European Patent 0367273.

[83] Q. Wei, L. Ping, W. Di, and X. Feng-kai, "A new sorting algorithm for radar emitter recognition," in *Computer, Mechatronics, Control and Electronic Engineering (CMCE), 2010 International Conference on*, vol. 5, pp. 407–410, IEEE, 2010.

[84] J. D. Parker, "Deinterleaver Technology for Future Electronic Support Measures (ESM) systems," tech. rep., DTIC Document, 1992.

[85] S. W. Driggs, T. J. Eckenrode, W. S. Richter, and J. L. Twoey, "Robust pulse deinterleaving," July 2010. US Patent 7,760,135.

[86] S. P. Drake, B. D. Anderson, and C. Yu, "Method and system for deinterleaving signals," Jan. 2012. US Patent 8,102,297.

[87] J. Roe, S. Cussons, and A. Feltham, "Knowledge-based signal processing for radar ESM systems," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 137, pp. 293–301, IET, 1990.

[88] N. Whittall, "Signal sorting in ESM systems," *Communications, Radar and Signal Processing, IEE Proceedings F*, vol. 132, no. 4, pp. 226–228, 1985.

[89] J. Anderson, "Radar signal categorization using a neural network.," *Neural Networks*, vol. 1, p. 422, 1988.

[90] V. Chandra, B. Jyotishi, and R. Bajpai, "Some new algorithms for ESM data processing," in *System Theory, 1988., Proceedings of the Twentieth Southeastern Symposium on*, pp. 108–112, IEEE, 1988.

[91] V. Chandra and R. Bajpai, "ESM data processing parametric deinterleaving approach," in *TENCON'92."Technology Enabling Tomorrow: Computers, Communications and Automation towards the 21st Century.'1992 IEEE Region 10 International Conference.*, pp. 26–30, IEEE, 1992.

[92] W. J. Baumann, "Pulse deinterleaving signal processor and method," May 1985. US Patent 4,516,220.

[93] R. Beale and T. Jackson, *Neural Computing-an introduction*. CRC Press, 1990.

[94] T. M. Eldos and A. S. Almazyad, "Adaptive resonance theory training parameters: pretty good sets," 2010.

[95] L. Rokach and O. Maimon, "Clustering methods," in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.

[96] G. Folino, A. Forestiero, and G. Spezzano, "Swarm-based distributed clustering in peer-to-peer systems," in *International Conference on Artificial Evolution (Evolution Artificielle)*, pp. 37–48, Springer, 2005.

[97] E. Kreyszig, *Advanced engineering mathematics*. John Wiley & Sons, 2006.

Appendix A

Derivations

A.1 PRI Analysis

Suppose the number of elements in RF sub-pattern P_h is J_h. And assume the number of frames that constitute the time sequence of the emitter is I. Hence, the s_{11h} based on equation (4.34) is given by;

$$s_{11h} = J_h I \tag{A.1}$$

The indexes of pulses within the cluster are n_h. While;

$$n_h = \{0, 1, ..., (J_h I - 1)\} \tag{A.2}$$

Hence;

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \sum_{n=0}^{s_{11h}-1} (n_{h1} \cup n_{hj} \cup ... \cup n_{hJ_h})^2 \tag{A.3}$$

where, n_{hj} is the sample indexes of time sub-sequence C_{hj} with respect to the time-sequence of the cluster h. Hence;

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \frac{1}{6} J_h I \left(J_h I - 1\right) \left(2 J_h I - 1\right) \tag{A.4}$$

or

$$s_{22h} = \sum_{n=0}^{s_{11h}-1} (n_h)^2 = \frac{1}{6} s_{11h} \left(s_{11h} - 1\right) \left(2 s_{11h} - 1\right) \tag{A.5}$$

y1:

$$\sum_{n=0}^{s_{11h}-1} t_h(n) = \sum_{n=0}^{s_{11h}-1} (C_{h1} \cup C_{hj} \cup ... \cup C_{hJ_h}) \tag{A.6}$$

$$\sum_{n=0}^{s_{11h}-1} (C_{h1} \cup C_{hj} \cup ... \cup C_{hJ_h}) = \sum_{i=0}^{I-1} C_{h1}(i) + ... \sum_{i=0}^{I-1} C_{hj}(i) + .. + ... \sum_{i=0}^{I-1} C_{hJ_h}(i) \tag{A.7}$$

$$\sum_{n=0}^{s_{11h}-1} t_h(n) = \sum_{j=1}^{J_h} \sum_{i=0}^{I-1} C_{hj} \tag{A.8}$$

$$\sum_{i=0}^{I-1} C_{hj}(i) = \sum_{i=0}^{I-1} K \cdot T \cdot (i-1) + P_{hj} + t_o \tag{A.9}$$

$$\sum_{i=0}^{I-1} C_{hj}(i) = K \cdot T \cdot \sum_{i=0}^{I-1} \cdot (i-1) + T \sum_{i=0}^{I-1} P_{hj} + \sum_{i=0}^{I-1} t_o \tag{A.10}$$

$$\sum_{i=0}^{I-1} (i-1) = \frac{1}{2} I(I-3) \tag{A.11}$$

$$\sum_{j=1}^{J_h} \sum_{i=0}^{I-1} C_{hj} = \frac{K \cdot T}{2} \cdot \sum_{j=1}^{J_h} I(I-3) + T \sum_{j=1}^{J_h} \sum_{i=0}^{I-1} P_{hj} + t_o I J_h \tag{A.12}$$

$$= \frac{K \cdot T}{2} \cdot J_h I(I-3) + T \sum_{j=1}^{J_h} \sum_{i=0}^{I-1} P_{hj} + t_o I J_h \tag{A.13}$$

Let

$$\upsilon_h = \sum_{j=1}^{J_h} P_{hj} \tag{A.14}$$

$$\sum_{i=0}^{I-1} \sum_{j=1}^{J_h} P_{hj} = \sum_{i=0}^{I-1} \upsilon_h = I \cdot \upsilon_h \tag{A.15}$$

hence;

y1:

$$\sum_{n=0}^{s_{11h}-1} t_h(n) = \sum_{j=1}^{J_h} \sum_{i=0}^{I-1} C_{hj} \tag{A.16}$$

$$= \frac{K \cdot T}{2} \cdot J_h I(I-3) + I \cdot \upsilon_h + t_o I J_h \tag{A.17}$$

$$= \frac{K \cdot T}{2} \cdot J_h s_{11h} \left(s_{11h} - 3\right) + TI \cdot \upsilon_h + t_o s_{11h} J_h \tag{A.18}$$

Assuming $t_o = 0$,

$$\sum_{n=0}^{s_{11h}-1} t_h \left(n\right) = T \left(\frac{K}{2} \cdot J_h s_{11h} \left(s_{11h} - 3\right) + I \cdot \upsilon_h\right) \tag{A.19}$$

$$y_1 = T \left(\frac{1}{2} IK J_h^2 \left(IJ_h - 3\right) + A\upsilon_h\right) \tag{A.20}$$

---

y2:

$$y_2 = \sum_{n=0}^{s_{11h}-1} n_h \cdot t_h \left(n\right) \tag{A.21}$$

$$y_2 = \sum_{j=1}^{J_h} \sum_{n=0}^{s_{11h}-1} n_{hj} \cdot C_{hj} \left(n\right) \tag{A.22}$$

Now, n_{hj} is given by;

$$n_{hj} \left(i\right) = J_h \cdot \left(i - 1\right) + j \tag{A.23}$$

$$\sum_{j=1}^{J_h} \sum_{n=0}^{I-1} n_{hj} \left(n\right) \cdot C_{hj} \left(n\right) \tag{A.24}$$

$$= \sum_{j=1}^{J_h} \sum_{n=0}^{I-1} \left(J_h \cdot \left(n - 1\right) + j\right) \cdot \left(K \cdot T \cdot \left(n - 1\right) + T \cdot P_{hj} + t_o\right) \tag{A.25}$$

$$= \sum_{j=1}^{J_h} \sum_{n=0}^{I-1} KTJ_h n^2 + \left(jKT - 2KTJ_h + TJ_h P_{hj} + J_h t_o\right) n + \left(KTJ_h - jKT + jTP_{hj} - TJ_h P_{hj} + jt_o - J_h t_o\right) \tag{A.26}$$

Performing summation over n only;

$$y_2 = \sum_{j=1}^{J_h} \left(\frac{1}{2} I^2 KT - \frac{3}{2} IKT + It_o \right) j +$$
$$+ \frac{1}{6} \sum_{j=1}^{J_h} \left(13KTI - 9KTI^2 + 3I^2 t_o + 2KTI^3 + 9It_o \right) \quad J_h +$$
$$+ \sum_{j=1}^{J_h} \left(\frac{1}{2} I^2 TJ_h - \frac{3}{2} ITJ_h \right) P_{hj}$$
$$+ \sum_{j=1}^{J_h} (IT)\, jP_{hj} \tag{A.27}$$

$$y_2 = \tfrac{1}{12} KT(-9IJ_h + 3I^2 J_h + 17IJ_h^2 - 15I^2 J_h^2 + 4I^3 J_h^2) \tag{A.28}$$
$$+ \tfrac{1}{12} \left(6IJ_h t_o - 12IJ_h^2 t_o + 6I^2 J_h^2 t_o \right)$$
$$+ IT \textstyle\sum_{j=1}^{J_h} jP_{hj} + \left(\frac{1}{2} I^2 TJ_h - \frac{3}{2} ITJ_h \right) \sum_{j=1}^{J_h} P_{hj} \tag{A.29}$$

Note that,the size of sub-sequences, their number of elements, and their elements values are unknown. Hence, the terms $\sum_{j=1}^{J_h} P_{hj}$, and $\sum_{j=1}^{J_h} jP_{hj}$ can not explicitly evaluated. In order to give an example for calculating these two terms let us take this case;

$P = \{F_2, F_1, F_1, F_1, F_2, F_3\}$. Hence;

$P_1 = \{1, 2, 3\}$

$P_2 = \{0, 4\}$

$P_3 = \{5\}$

$\sum_{j=1}^{J_h=3} P_{1j} = 1 + 2 + 3$

$\sum_{j=1}^{J_h=2} P_{2j} = 0 + 4$

Back to our derivation, let $\upsilon_h = \sum_{j=1}^{J_h} P_{hj}$, and $\rho_h = \sum_{j=1}^{J_h} jP_{hj}$

Assuming $t_o = 0$, then equation (A.28) becomes;

$$y_2 = \tfrac{1}{12} KT(-9IJ_h + 3I^2 J_h + 17IJ_h^2 - 15I^2 J_h^2 + 4I^3 J_h^2)$$
$$+ IT \textstyle\sum_{j=1}^{J_h} jP_{hj} + \left(\frac{1}{2} I^2 TJ_h - \frac{3}{2} ITJ_h \right) \sum_{j=1}^{J_h} P_{hj} \tag{A.30}$$

$$\begin{aligned} &= \frac{1}{12}KT(-9IJ_h + 3I^2J_h + 17IJ_h^2 - 15I^2J_h^2 + 4I^3J_h^2) \\ +IT\rho_h &+ \left(\frac{1}{2}I^2TJ_h - \frac{3}{2}ITJ_h\right)\upsilon_h \end{aligned} \tag{A.31}$$

$$y_2 = -\frac{3}{4}IKTJ_h^2 + \frac{17}{12}IKTJ_h^3 + \frac{1}{4}I^2KTJ_h^3 - \frac{5}{4}I^2KTJ_h^4 + \frac{1}{3}I^3KTJ_h^5 + ITJ_h\rho_h - \frac{3}{2}ITJ_h^2\upsilon_h + \frac{1}{2}I^2TJ_h^3\upsilon_h \tag{A.32}$$

$$s_{12h} = \sum_{n=0}^{s_{11h}-1} n = \frac{1}{2}J_hI\left(J_hI - 1\right) \tag{A.33}$$

$$T_h = \frac{1}{s_{11h}s_{22h} - s_{12h}^2}\left[s_{11h}\sum_{n=0}^{s_{11h}-1} n \cdot t_h\left(n\right) - s_{12h}\sum_{n=0}^{s_{11h}-1} t_h\left(n\right)\right] \tag{A.34}$$

$$\begin{aligned} T_h &= \frac{\left(-9KTJ_h + 3IKTJ_h + 8KTJ_h^2 - 15IKTJ_h^2 + 4I^2KTJ_h^2 + 12IKTJ_h^3 - 3I^2KTJ_h^4\right)}{J_h\left(I^2J_h^2 - 1\right)} \\ &+ \frac{12T\rho_h + 6T\upsilon_h - 18TJ_h\upsilon_h}{J_h\left(I^2J_h^2 - 1\right)} \end{aligned} \tag{A.35}$$

Removing 1 from denominator because it is insignificant gives;

$$\begin{aligned} T_h &= \frac{\left(-9KTJ_h + 3IKTJ_h + 8KTJ_h^2 - 15IKTJ_h^2 + 4I^2KTJ_h^2 + 12IKTJ_h^3 - 3I^2KTJ_h^4\right)}{J_h\left(I^2J_h^2\right)} \\ &+ \frac{12T\rho_h + 6T\upsilon_h - 18TJ_h\upsilon_h}{J_h\left(I^2J_h^2\right)} \end{aligned} \tag{A.36}$$

$$T_h = KT\left(\frac{17}{I^2} + \frac{15}{I} - \frac{18}{I^2J_h} - 3J_h - \frac{15J_h}{I} + 4J_h^2\right) + T\left(\frac{12\rho_h}{I^2J_h^3} + \frac{\left(6 - 18J_h\right)\upsilon_h}{I^2J_h^3}\right) \tag{A.37}$$

$$T_h = KT\left(\frac{15}{I} - 3J_h - \frac{15J_h}{I} + 4J_h^2\right) \tag{A.38}$$

When I is large, the term $\frac{12\rho_h}{I^2 J_h^3}$ becomes insignificant because $I^2 J_h^3$ is very large compared to $12\rho_h$. The same concept applied to term υ_h, hence;

$$T_h = KT\left(\frac{15}{I} - 3J_h - \frac{15J_h}{I} + 4J_h^2\right) \tag{A.39}$$

Assume cluster A and cluster B are two different clusters that belong to the same emitter. Assume the values of J_h for the the two clusters are respectively $J_A = a$, and $J_B = b$. Let us find the ratio of estimated parameters T_h of both clusters as follows;

$$R = \frac{KT\left(\frac{15}{I} - 3a - \frac{15a}{I} + 4a^2\right)}{KT\left(\frac{15}{I} - 3b - \frac{15J_h}{I} + 4b^2\right)} \tag{A.40}$$

$$R = \frac{15 - 15a - 3I\,a + 4I\,a^2}{15 - 15b - 3I\,b + 4I\,b^2} \tag{A.41}$$

For realistic emitter scenario, let us assume that, for two different clusters of an emitter, the maximum value for the ratio $\frac{J_a}{J_b}$ is 4.

Now, based on $\frac{J_a}{J_b}$ condition, the possible cases are presented in table. The table shows the value R_{dB} (i.e. $10\,log\,(R)$) of presented cases. The value of I that was used in calculation is 200. The values in the table are almost fixed for a given range of I. For instance, when the value of I drops down to 50, then the values in table change by only very small fraction.