EMITTER IDENTIFICATION TECHNIQUES
IN ELECTRONIC WARFARE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MEHMET KADİR ASLAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2006

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. İsmet ERKMEN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Zafer ÜNVER
Supervisor

Examining Committee Members

Prof. Dr. Mete SEVERCAN (METU, EEE)

Prof. Dr. Zafer ÜNVER (METU, EEE)

Prof. Dr. Kemal LEBLEBİCİOĞLU (METU, EEE)

Assoc. Prof. Dr. Sencer KOÇ (METU, EEE)

Levent ALKIŞLAR (M. Sc.) (ASELSAN)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : MEHMET KADİR ASLAN

Signature :

ABSTRACT

EMITTER IDENTIFICATION TECHNIQUES
IN ELECTRONIC WARFARE

ASLAN, Mehmet Kadir
M.Sc., Department of Electrical and Electronics Engineering

Supervisor: Prof. Dr. Zafer ÜNVER

September 2006, 92 pages

In this study, emitter identification techniques have been investigated and a schema has been proposed to solve the emitter identification problem in Electronic Warfare systems.

Clustering technique, histogram based deinterleaving techniques and a continuous wavelet transform based deinterleaving technique have been reviewed. A receiver simulator software has been developed to test the performance of these techniques and to compare them against each other. To compensate the disadvantages of these techniques, a schema utilizing the beneficial points of them has been developed. With the modifications proposed a resultant schema has been obtained.

Proposed schema uses clustering and deinterleaving together with other proposed modifications. Tests made through out this study have shown that this usage improves performance of emitter identification system. Hence, proposed schema can be used to identify the emitters in real EW systems.

Keywords: Electronic Warfare, Emitter Identification, Clustering, Deinterleaving, Wavelet Transform.

ÖZ

ELEKTRONİK HARPTE
YAYIN TANIMLAMA TEKNİKLERİ

ASLAN, Mehmet Kadir
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Zafer ÜNVER

Eylül 2006, 92 sayfa

Bu çalışmada yayın tanımlama teknikleri incelenmiş ve Elektronik Harp sistemlerindeki yayın tanımlama problemini çözmek için bir yöntem önerilmiştir.

Kümeleme tekniği, histogram tabanlı ayrıştırma teknikleri ve sürekli dalgacık dönüşümü tabanlı bir ayrıştırma tekniği incelenmiştir. Bu tekniklerin performansını ölçmek ve bu algoritmaları kendi aralarında karşılaştırmak için bir almaç simülatörü yazılımı geliştirilmiştir. Bu tekniklerin dezavantajlarının etkisini azaltmak için, bu tekniklerin bir kısmının birleşiminden oluşan bir yöntem geliştirilmiştir. Yapılan iyileştirmelerle yöntem son halini almıştır.

Önerilen yöntem kümeleme ve ayrıştırma tekniklerini diğer önerilen değişikliklerle birlikte kullanmaktadır. Bu çalışma sırasında yapılan testler bu kullanımın yayın tanımlamanın performansını arttırdığını göstermektedir. Önerilen yöntem gerçek Elektronik Harp sistemlerinde yayın tanımlama için kullanılabilir.

Anahtar Kelimeler: Elektronik Harp, Yayın Tanımlama, Kümeleme, Ayrıştırma, Dalgacık Dönüşümü

To My Family

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my supervisor Prof. Dr. Zafer ÜNVER for his supervision and support throughout this thesis study.

I am also grateful to ASELSAN Inc. for the sources they let me to use.

I want to thank to my colleagues who had inspired me intentionally or unintentionally to tidy up the ideas in my mind. I also want to thank my friends for their great morale support during this thesis.

Last, but not the least I would like to thank my love, Gözde, for giving me the strength to complete this thesis and for sympathy during this study. This thesis would not be completed without her encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

ABBREVIATIONS

A-DIF	:	All DIFferences
AOA	:	Angle Of Arrival
ASP	:	Antenna Scan Period
AST	:	Antenna Scan Type
BW	:	Beam Width
C-DIF	:	Cumulative DIFferences
dBm	:	deci Bell milli
DTOA	:	Difference TOA
ECCM	:	Electronic Counter Counter Measures
ECM	:	Electronic Counter Measures
ELINT	:	ELectronic INTelligence
ESM	:	Electronic Support Measures
EW	:	Electronic Warfare
Hz	:	Hertz
MATLAB	:	MATrix LABoratory
MOP	:	Modulation On Pulse
MTI	:	Moving Target Indicator
PA	:	Pulse Amplitude
PDW	:	Pulse Descriptor Word
PRF	:	Pulse Repetition Frequency
PRI	:	Pulse Repetition Interval
PW	:	Pulse Width
RWR	:	Radar Warning Receiver
S-DIF	:	Sequential DIFferences
SNR	:	Signal to Noise Ratio
TOA	:	Time Of Arrival

CHAPTER 1

INTRODUCTION

The ultimate goal of passive Electronic Warfare (EW) systems is to classify radar signals. Classification of radars is possible by using their unique characteristics. Some of these characteristics may be directly measured and some other must be derived from measured parameters. Once the characteristics of radars are determined they can be identified.

Usually derivation of parameters is not easy because pulse trains from a number of different sources are received on the same communication channel. Assuming that each of these trains has different characteristics, one may be interested in sorting out these pulse trains, thus identifying which pulse train comes from which source. This task is termed as deinterleaving [1].

The task of deinterleaving is commonly found in applications such as radar detection and electronic warfare. Main aim of the deinterleaving process is to detect and extract repeating structures. Then the period of this structure and the type of periodicity can be examined [1].

1.1. ELECTRONIC WARFARE

The concept of Electronic Warfare is to use electromagnetic spectrum to determine enemy's order of battle, intensions and capabilities or prevent hostile use of electromagnetic spectrum [1]. Passive EW systems monitor the electromagnetic spectrum by measuring the parameters of intercepted signals, and use these data to identify the intercepted emitter.

Basically EW systems are Electronic Intelligence (ELINT), Electronic Support Measures (ESM), Radar Warning Receiver (RWR) and Electronic Counter Measures (ECM) systems. ELINT systems provide a form of detailed measurement for signal analysis, whereas ESM systems provide a real time projection of radar activity. The major difference between ELINT and ESM systems is their response time. RWRs indicate whether the platform is under attack. RWR may be part of an active system which applies ECM to a hostile system.

Since this subject is closely related to the military applications, practically there are only a few detailed documents in public literature. If we also think that the radar systems are getting more complicated to prevent themselves from being detected, the importance of deinterleaving for ESM and ELINT systems increases.

The goal of deinterleaving is to classify radar signals by their unique characteristics and use this data to [10]:

- identify enemy radars operating in the environment,
- determine their location or direction,
- inform friendly forces about their threats,
- display this information to the operator.

A simplified block scheme of a typical EW system is given in **Figure 1.1**.

In radar signal processing, processing time is an important matter. Typical pulse repetition rates range from 50 Hz to 500 kHz, which means that EW systems should be capable of processing 500k pulses per second. Since the pulse repetition frequency resolution required is not the same for the whole band, a multi-resolution approach seems to be convenient [9].

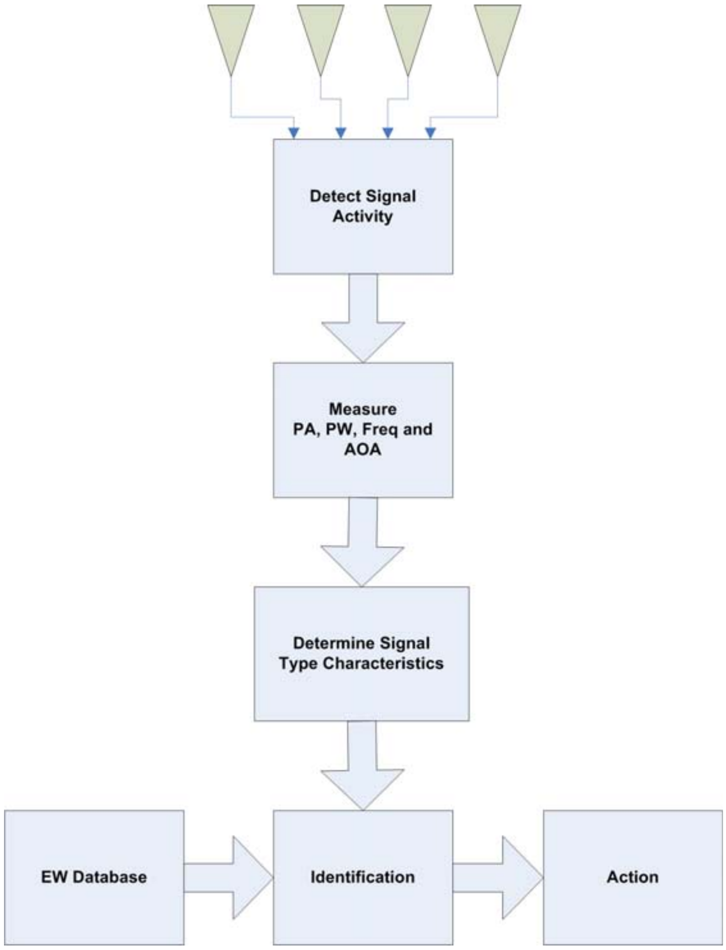

Figure 1.1 Signal Processing Steps of EW Systems (Simplified)

A system used to detect and identify radars will measure certain characteristics of the radar emissions to find out the nature of the source [3]. The majority of radars that are encountered in practice emit energy in predefined sequences of pulses. The measured parameters are generally Pulse Amplitude (PA), Pulse Width (PW), Pulse Frequency, Pulse Direction (Angle of Arrival, AOA) and Time of Arrival (TOA). The timing of pulses may follow simple or complex patterns described by measurable parameters [2]. Furthermore, an ESM system can derive Pulse Repetition Interval (PRI), Antenna Scan Period (ASP), Antenna Scan Type (AST) and Beam

Width (BW) parameters from these measured parameters. Usage of these parameters for deinterleaving and emitter identification is given in **Table 1.1**.

Table 1.1 Usage of Emitter Parameters in Deinterleaving and Identification

		Pulse Sequence Deinterleaving	**Emitter Identification**
Measured	**Frequency**	Very Useful	Very Useful
	Angle of Arrival	Very Useful	Not Useful
	Pulse Width	Very Useful	Some Use
	Pulse Amplitude	Some Use	Not Useful
	Time of Arrival	Very Useful	Not Useful
Derived	**Pulse Repetition Interval**	Very Useful	Very Useful
	PRI Type	Very Useful	Very Useful
	Antenna Scan Period and Type	Not Useful	Very Useful
	Beam Width / Lobe Duration	Not Useful	Very Useful

Pulse characteristics of radars are different because of their specifications, missions, Electronic Counter Counter Measure (ECCM) capabilities and design technologies. A radar system may intentionally vary one or more of its parameters to prevent them from being detected or to improve their functionality. This variation may be from pulse to pulse or from pulse group to pulse group.

In the following sub sections brief descriptions of these parameters are given.

1.1.1. Pulse Amplitude

Majority of radars use directive antennas making a mechanical rotation, and hence the pulse amplitude received from a radar changes in time. The pulse amplitude will be high whenever the main beam of radar is directed to the receiving system. When pulses from side or back lobe of radar are being received, pulses may not be detectable or the pulse amplitude may be very low. Detection of a pulse is directly related to the sensitivity of the receiver.

1.1.2. Pulse Width

Pulse width of the radar determines radar's resolution. Radars rarely change their pulse width to take precaution to be detected. But radars can change their pulse width in accordance with the PRI parameter to improve their search capability. The pulse width measurement accuracy is related to pulse amplitude. For small pulse amplitudes the pulse width may be incorrect due to low Signal to Noise Ratio (SNR).

1.1.3. Frequency

Radars usually change their frequency to raise difficulty of being detected and being jammed. The change in frequency can be random or deterministic and can be from pulse to pulse or group to group.

1.1.4. Angle of Arrival

Since a radar cannot change its location very rapidly, all pulses received from the radar will be received from the same direction. Even if a radar or a receiving system is moving, this movement will be very slow as compared to the pulse rate. Thus the angle of arrival parameter will be very useful in deinterleaving. If the receiving system is moving, by use of angle of arrival changes in a relatively long time, location of the radar can be estimated [1].

1.1.5. Time of Arrival

A measured time of arrival parameter is related to the PRI parameter of radars. Radars can use more than one PRI and can change their PRI values in very different types.

PRI determines radar's maximum unambiguous range and maximum unambiguous velocity [1]. Radars may change their PRI modulation to resolve ambiguities or to improve their ECCM capabilities. Mostly encountered PRI types are given in the paragraphs below.

1.1.5.1. Constant PRI

A radar is said to have a constant PRI if peak variations in PRI are less than 1% of the mean PRI. These variances are assumed to be incidental. If the average PRI value is to be used to estimate the radar's maximum unambiguous range, an accuracy of 1% is sufficient.

1.1.5.2. Jittered PRI

Jittered PRI is large variations in PRI up to 30% of the mean PRI. Such variations are generally used for ECCM. In fact the detection of jittered PRI may be very difficult for EW systems.

1.1.5.3. Dwell & Switch PRI

In this type of radar, several different PRIs are selected for use. The radar transmits pulses at a constant PRI for a dwell time and switches its PRI for the next dwell time. This technique is used to resolve range ambiguities in pulse Doppler radars [1].

1.1.5.4. Staggered PRI

Staggered PRI means use of two or more PRIs in a fixed sequence. Roughly, it can be said that PRI is changed from pulse to pulse. The staggered PRI is generally used to eliminate blind speeds in Moving Target Indicator (MTI) systems [1].

1.1.5.5. Sliding PRI & Sinusoidal PRI

Sliding PRI is characterized by monotonically increasing or decreasing PRI followed by a rapid jump to one extreme limit when the other extreme limit is reached. A sinusoidal variation in PRI can also improve radar's functionality. Sliding and sinusoidal PRIs eliminate blind ranges [1].

1.1.5.6. Pulse Group

Radars may transmit groups of closely spaced pulses separated by longer time intervals. This type of variation may be used to resolve range ambiguity and velocity ambiguity problems together.

1.1.6. Deinterleaving Problem

For the purpose of detecting and identifying radars in the environment, the pulse sequences received from radars are used. The problem of determining the presence of a specific emitter in the environment is a problem of detecting a consistent pulse sequence in the incoming stream of interleaved pulses [9].

A significant problem arises when the signals from two or more emitters overlap in time. Pulses arrive at the receiver of an EW system in natural time order and so become interleaved. This complicates the task of identifying individual sequences. It is sometimes difficult or even impossible to determine which pulse belongs to which emitter, or how many emitters are present in an interleaved pulse stream.

To express this problem more mathematically, let us consider N signal sources (N is unknown), each one of which generates a periodic pulse train with period T_i where i is ranging from 1 to N [9]. Assuming that the pulses contain no amplitude or width information that can be used for source identification (evaluation of amplitude, width and frequency information can be combined later); the period becomes the only distinguishing feature. These received pulses are affected by noise and interleaved in a single communication channel.

If we denote the finite basic pulse used in all trains by *g(t)*, then the received signal from the i^{th} source can be written as,

$$s_i(t) = \sum_{n=-\infty}^{\infty} g(t - nT_i - \tau_i) \qquad \textbf{(1.1)}$$

where τ_i is the reference time for this signal.

The global interleaved received signal is then expressed by,

$$s(t) = \sum_{i=1}^{N} s_i(t) + n(t) \qquad \textbf{(1.2)}$$

where *n(t)* is the observation noise.

The problem is to determine all periods T_i once *s(t)* is given. The deinterleaving problem can be stated as a combination of two sub problems. The first problem is *period estimation* which is focusing on the estimation of the period parameters for source identification. The second problem is *signal retrieval* where pulses corresponding to each source are recovered [9]. This recovery is often necessary if one desires to generate synchronous versions of the incoming signal. For both cases, the only source of information is the knowledge of the noisy global signal observed during a time period.

1.2. DEINTERLEAVING TECHNIQUES

In the literature there are known techniques on the deinterleaving problem. Each technique may have some limitations and some disadvantages. A deinterleaving algorithm must be robust against missing pulses and timing jitter, and must be sensitive to simple, staggered and complex pulse sequences. Conditions that have impact on pulse train deinterleaving are [11]:

- Pulse overlap,

- Dropped pulses,
- Extraneous pulses (multipath),
- Intermittent pulse trains (effect of radar's scan characteristic),
- Pulse shadowing,
- False alarms,
- Receiver blanking.

Techniques applied to the problem of deinterleaving are introduced in the following sections.

1.2.1. Cluster Analysis

Clustering is a method by which the parameters characterizing an object are so grouped that objects in a group are more strongly related to each other than those in different groups.

In the literature there are many algorithms for clustering. Most of these algorithms are applied to the emitter identification problem [12], [13]. But use of a clustering algorithm alone may not be sufficient for full deinterleaving of all emitters.

A simple deinterleaving method utilizing clustering algorithms may use some or all of frequency, AOA and PW parameters. Using a proper distance function with the clustering algorithm, pulses with different parameters can be deinterleaved very simply and very quickly. But when two radars located in the same direction are emitting pulses at the same frequency and the same pulse width, they cannot be deinterleaved with this method. In this case deinterleaving is only possible by using TOA information. The clustering method may be used in combination with another method using TOA and PA information to resolve this problem.

In the next chapter one of the clustering algorithms is implemented to see the performance and ability of clustering for deinterleaving.

1.2.2. TOA Deinterleaving Methods

Another type of deinterleaving techniques is based on the TOA parameter because earlier EW systems were only capable of measuring TOA and frequency.

An established procedure for processing of interleaved pulse sequences is based on difference time of arrival (DTOA) histogram [1]. The essence of this method is first to make a coarse search in PRI using the DTOA histogram, removing any sequences with the detected PRIs from the interleaved pulse stream, and then to repeat the process on the remaining data. A number of complications impact the effectiveness of this approach [7]. The DTOA histogram:

- produces false detections at multiples of the true PRIs,
- is sensitive to the bin width,
- fails to detect complex pulse sequences (sequences changing PRI in time),
- is computation-intensive procedure,
- interleaved pulse streams produce a background of false pulse gaps in the histogram, which bury the true PRIs.

Another approach is to search a periodogram for pulse repetition frequencies (PRF) [8]. The periodogram method has some disadvantages [9]. The periodogram:

- produces false detections at multiples of the true PRF,
- fails to detect short sequences,
- fails to detect complex sequences,
- provides an inefficient search strategy.

Another method is the sequence search method [6]. This method starts from an initial PRI and tries to find following pulses at that PRI. If pulses can not be found, then it starts the pulse search with another PRI value. It is not difficult to guess that this method is not much effective in this form. But once PRI is known, this method can retrieve the correct pulses very simply.

A relatively new technique for deinterleaving is to use a wavelet based detector [9]. This method is based on a continuous wavelet transform used for detecting radar pulse sequences. It operates directly on interleaved pulse streams. The detector function contains two arguments: the one corresponding to location in time, and the other corresponding to a characteristic period or scale *T*. The detector function detects whether if there exists an emitter at time *t* with period *T*. A single adjustable parameter M designates the resolution of the detector. The new method overcomes the following complications encountered in other approaches [9]:

- harmonics are suppressed by a factor of four or more,
- cross interference effects between interleaved pulse sequences are minimized,
- detection of complex sequences is improved,
- the procedure of searching for *T* is fast,
- robust against missing pulses and timing jitter.

In the next chapter some of these techniques are implemented to see their performance and abilities. Thus it will be possible to compare these techniques with each other.

1.3. SCOPE OF THE THESIS

In this thesis, the aim is to develop a schema for deinterleaving of radar pulse sequences. This schema has to reduce the disadvantages of the known techniques, without diminishing their capabilities.

In this scope, a schema which is originated from the known techniques in the literature is proposed for the deinterleaving task. This schema uses a clustering method and a wavelet detector method for PRI analysis. Also a sequence search method using both TOA and PA parameters will be proposed to improve the reliability of the schema. The proposed schema provides a complete method for deinterleaving and brings an improvement to other established approaches.

1.4. OUTLINE OF THE THESIS

The thesis consists of five chapters.

In Chapter 2, the clustering algorithm and various deinterleaving algorithms are studied. The clustering algorithm studied is the Improved Chain Algorithm with a normalized Euclidean distance. The flowcharts and improvements for this algorithm are given. TOA based deinterleaving algorithms, namely the Histogram Based Deinterleaving Algorithms and the Continuous Wavelet Based Deinterleaving Algorithm are studied. Flowcharts for these algorithms are also given. Some modifications proposed to these algorithms are explained.

In Chapter 3, the developed simulator and implementations of algorithms are explained. The details of the tests performed to observe the performance and ability of the algorithms are given.

In Chapter 4, the algorithms are combined to produce new deinterleaving schemas. Nine different schemas are generated and tested; and then the results are presented.

In Chapter 5, a conclusion for the thesis and comments for future works are given.

CHAPTER 2

PREVIOUS APPROACHES TO DEINTERLEAVING

The problem of emitter identification is a former problem in EW systems. Early studies on this problem use stationary parameters of emitters like AOA, frequency and PW with clustering techniques. In these methods mostly direction and frequency of the emitter are obtained. By collecting pulses coming at that direction and at that frequency, other parameters of emitter can be calculated. This information gives a chance for identifying the source of that emission.

Other approaches to this problem use TOA values of received pulses. These techniques try to estimate the PRI of emitter. Then using the estimated PRI pulses corresponding to that emitter can be retrieved and other parameters can be calculated using retrieved pulses.

Clustering approaches fail if there are two emitters located at the same direction and working at similar stationary parameters. On the other hand, TOA deinterleaving techniques may fail if there are many emitters working in the environment. In this chapter these techniques are explained in detail.

2.1. CLUSTERING TECHNIQUES

Clustering techniques try to group entities in a way that those entities belonging to a cluster have similarities and entities not in the same cluster are somehow different.

In the literature there are many algorithms for solving the clustering problem [4], [5]. These algorithms use different techniques but actually all of them try to achieve the same result. Performance and ability of algorithms depend on the problem type they are applied.

Formerly clustering algorithms are applied to the problem of source identification in Electronic Warfare [12] [13]. The aim of this study is neither to discuss the result of different clustering algorithms nor to measure their performance of these algorithms. Only one example of these algorithms is chosen for implementation to see the ability of clustering techniques for the emitter identification task.

2.1.1. Distance Function

A clustering algorithm needs a measure to determine how much an entity is similar to another entity. This measure is called a *distance metric*. A well known distance metric is the Euclidian distance given by,

$$d(\bar{x}_i, \bar{x}_j) = \sqrt{\sum_{k=1}^{N} \left| x_{ik} - x_{jk} \right|^2} \qquad \textbf{(2.1)}$$

where $\bar{x}_i$ and $\bar{x}_j$ are the entities distance of which is being measured and x_{ik} and x_{jk} are k^{th} feature of $\bar{x}_i$ and $\bar{x}_j$ respectively. Usually some normalization is employed to the test space in order to adjust the effect of each feature. Normalized Euclidian distance can be given by

$$d(\bar{x}_i, \bar{x}_j) = \sqrt{\sum_{k=1}^{N} \frac{\left| x_{ik} - x_{jk} \right|^2}{w_k^{\,2}}} \qquad \textbf{(2.2)}$$

where w_k is the weighting coefficient for the k^{th} feature.

2.1.2. Improved Chain Algorithm

A famous clustering algorithm is the Improved Chain Algorithm [4]. This algorithm first calculates the distance between the existing cluster centers and the current sample. Then the cluster center which has the smallest distance to the sample is found. If this distance is smaller than a certain threshold, this sample is put into that cluster. Otherwise, the sample becomes a new cluster center.

Cluster centers can be defined by the mean vector of the feature space. When a new entity is added to a cluster with *N* entities, the mean and the standard deviation of the cluster can be updated by the following equations:

$$\mu_{N+1} = \frac{N \cdot \mu_N + x_{N+1}}{N+1} \quad \textbf{(2.3)}$$

where μ_N is the mean for the *N* entities in the cluster, and x_{N+1} is the $N+1^{st}$ entity added to the cluster;

$$std_{N+1} = \sqrt{\frac{N}{N+1}(std_N{}^2 + \mu_N{}^2) - \mu_{N+1}{}^2 + \frac{1}{N+1}x_{N+1}{}^2} \quad \textbf{(2.4)}$$

where std_N is the standard deviation for the *N* entites in the cluster, and x_{N+1} is the $N+1^{st}$ entity added to the cluster.

2.1.3. Clustering Techniques in EW

Clustering techniques can be applied to the emitter identification problem by assuming each received pulse as an entity with the feature vector formed of three stationary parameters AOA, frequency and PW. Then an EW system can directly determine the emitter’s direction, mean frequency, mean pulse width and can indirectly calculate PRI, antenna scan type and antenna revolution period.

Using the Euclidian distance function with the clustering algorithm, pulses with different parameters can be deinterleaved very simply and very quickly. Weighting coefficients for the distance function are selected as given in **Table 2.1**. These values are related to the desired initial width of the cluster. With these weights, the threshold should be selected to be the square root of dimension of the feature space, which is 3 in this case.

Because of the varying nature of PA and TOA, they can not be used in clustering. If a receiver system can measure Modulation on Pulse (MOP) or Elevation Angle parameters, they can also be used as clustering parameters.

Table 2.1 Weighting Coefficients of Distance Function

Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**
16	4	2

When two radars located in the same direction are emitting pulses at the same frequency and the same pulse width, they can not be deinterleaved by the clustering method. However, to overcome this problem the clustering method may be used in combination with another method using TOA and/or PA information.

2.1.4. Modifications to Clustering Algorithm

Since there exists frequency or PW agile emitters, using a constant distance threshold may not be suitable. The following modifications improve the clustering algorithm for deinterleaving of agile emitters.

The algorithm starts with some expected weighting coefficients. The mean and standard deviation of cluster are calculated as soon as a new entity is added to the cluster. Once the standard deviation of the cluster is known, weighting coefficients for the distance metric can be adaptively updated. This update widens or narrows the width of the cluster.

Another modification to this algorithm may be merging of clusters. When pulses for a time period are clustered, clusters are checked. If any of them are overlapping, or very close to each other, then they are merged. Also the clusters having entities less than a predefined number are canceled.

The flowchart of this final algorithm is given in **Figure 2.1** and **Figure 2.2**.

2.2. HISTOGRAM BASED DEINTERLEAVING TECHNIQUES

One of the earlier studies on the emitter identification problem is histogram methods. Generally, DTOA histograms are used for determining the PRI. All DTOA histogram based methods use the data derived from TOA. According to the evaluation method of input TOA data used, the detection and decision method varies.

Generally, the TOA based deinterleaving is made in a successive manner. The process can be summarized as follows.

1. Take a block of pulses.
2. Eliminate pulses from the known emitters and update the emitter list.
3. Run the deinterleaving algorithm and find the most evident emitter.
4. Eliminate the pulses from this emitter and update the emitter list.
5. Repeat Steps 3 and 4 until the number of remaining pulses is less than a threshold value or no new emitter can be found.
6. Restart the process.

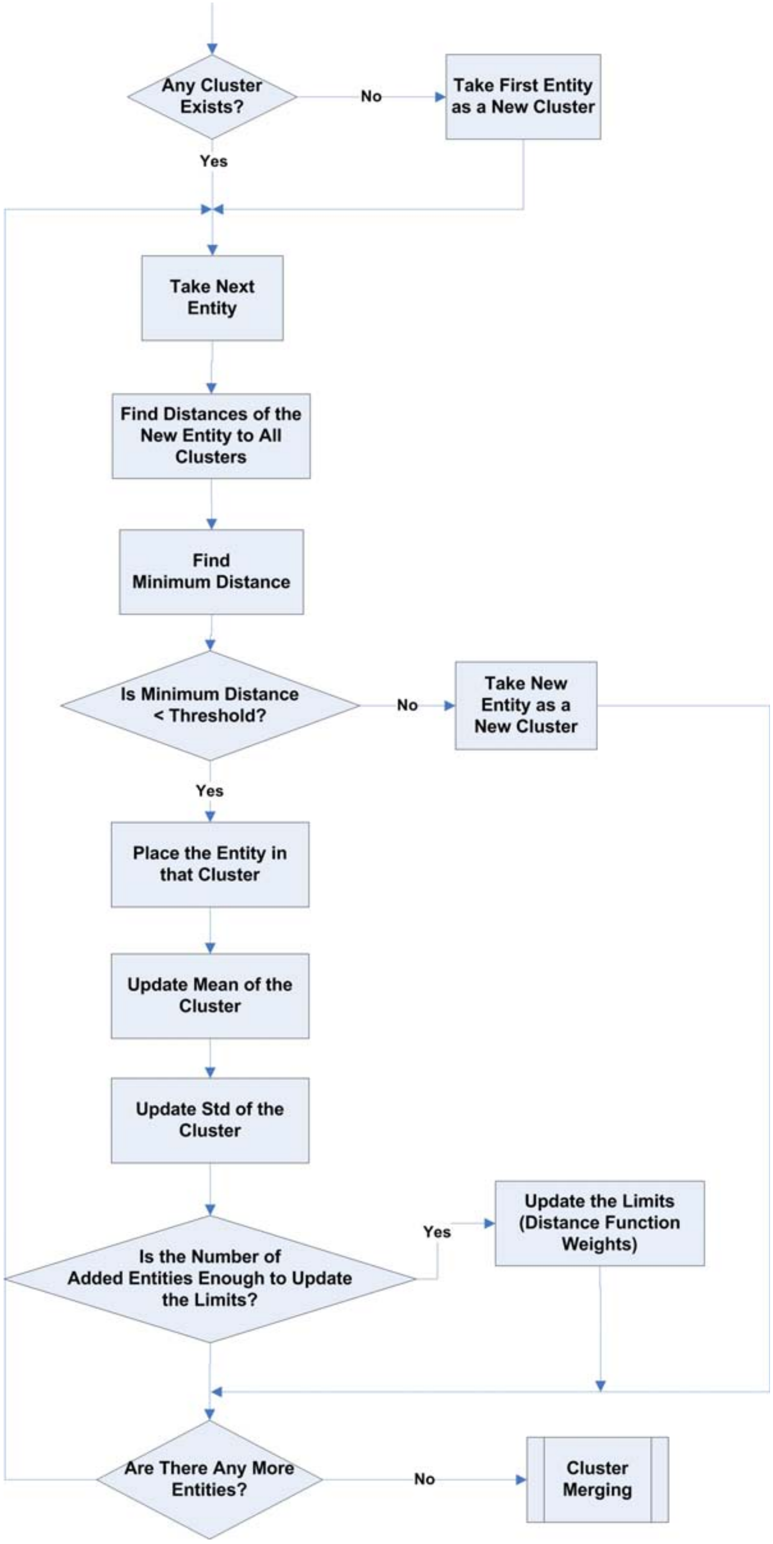

Figure 2.1 Improved Chain Clustering Algorithm

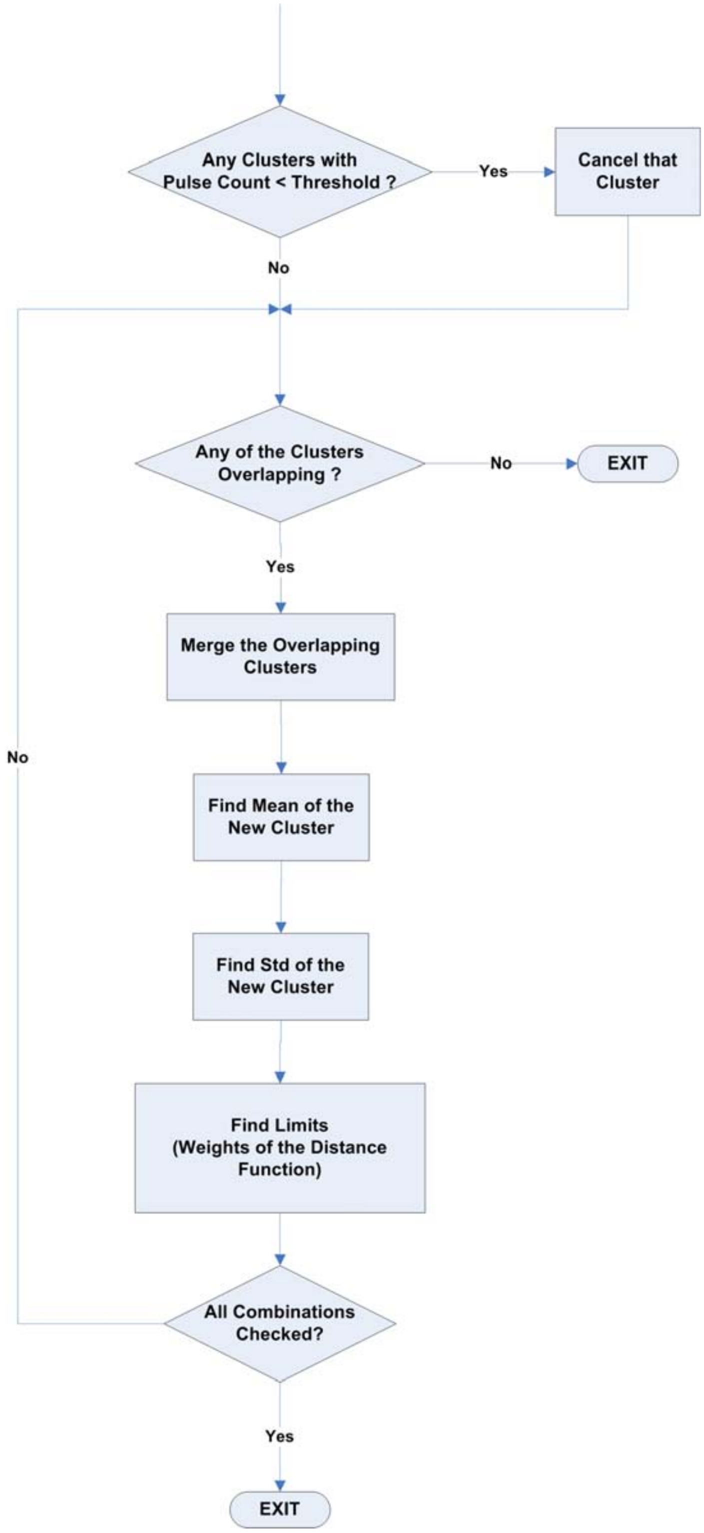

Figure 2.2 Cluster Merging Algorithm

An important point is the problem of managing data blocks. Data blocks must be processed as continuous time blocks. Time gaps in the pulse stream will produce higher differences, and this will directly affect the decision performance of the algorithm.

Several DTOA histogram based algorithms are examined in the following sections.

2.2.1. All Differences Histogram (A-DIF)

One former deinterleaving algorithm is based on the DTOA histogram of all consecutive pulses [1]. This method is also known as *Delta t* histogram. In this method a block of TOA data is taken, and all possible differences are calculated. However, in practice, only the differences up to a certain high level can be used. Then the histogram of this all differences data is generated. Starting from the first bin in the histogram a PRI estimate is taken, and a hypothesis test is applied. In practice the PRI estimate can be chosen as the first bin that exceeds a threshold. Using a threshold will improve the performance of the algorithm which is directly related to the allowed number of pulses that can describe an emitter. An example of this type of histogram is shown in **Figure 2.3**.

Since all differences are calculated, the histogram will have peaks at the true PRI and at its integer multiples. The hypothesis test checks the values these peaks. The histogram values at each integer multiple of the true PRI must be greater than a certain threshold level. The threshold values are related to the PRI and the number of expected pulses in that time interval. To decide that whether there is an emitter at that PRI, the first histogram peak must have at least

$$k \cdot \frac{T}{PRI}$$

pulses; the second histogram peak must have at least

$$k \cdot (\frac{T}{PRI} - 1)$$

pulses and so on. Here T is the total time interval of data at the histogram bin under the test; k is a number between 0 and 1, which is related to the percentage of missing pulses allowed.

If a PRI can not pass the hypothesis test, the algorithm is continued with the next PRI. If a PRI passes the hypothesis test, then pulses from that emitter are removed. (A method to remove pulses with a known PRI is given in the following sections.) Since some pulses are removed, the histogram is not valid anymore. Thus the algorithm is restarted by calculating the differences and reforming the histogram. The algorithm stops when no further detection occurs or number of pulses left is less then a predefined value. The flowchart of this algorithm is given **Figure 2.4**.

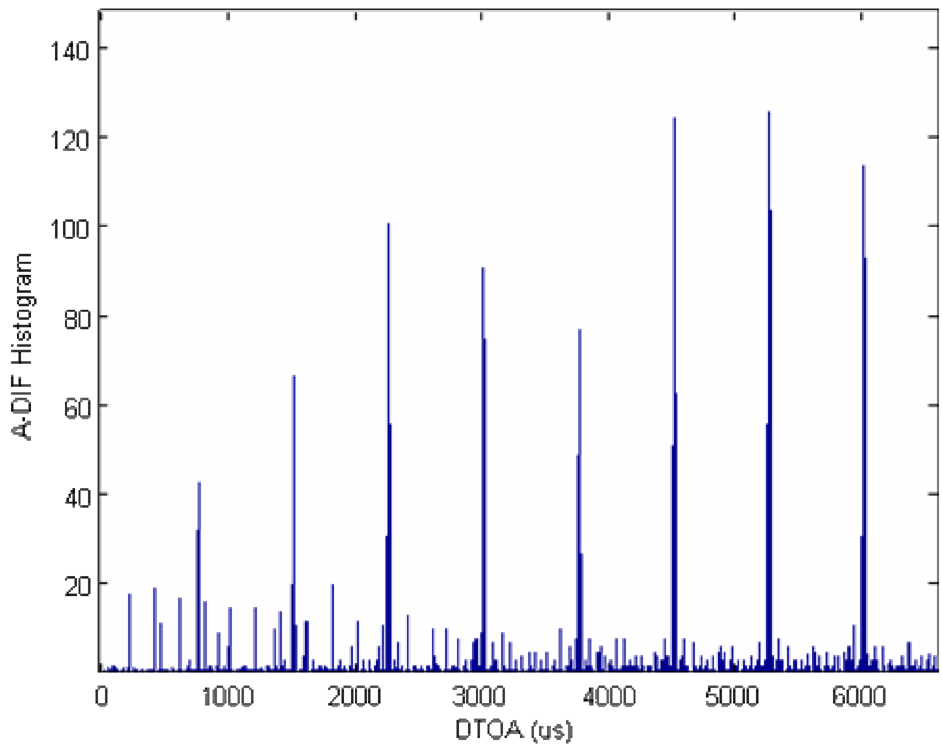

Figure 2.3 Example A-DIF Histogram

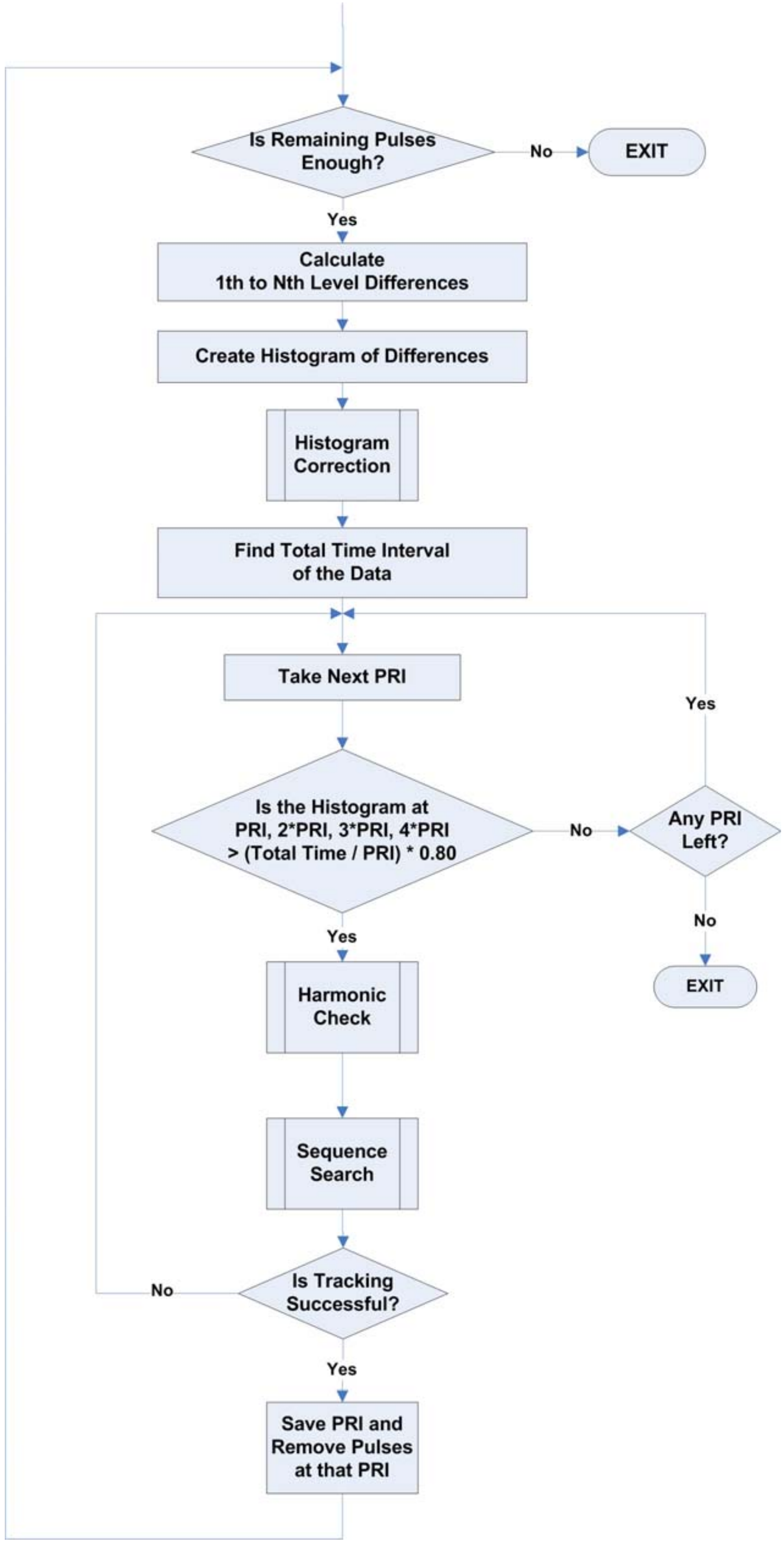

Figure 2.4 All Differences Histogram Deinterleaving Algorithm

2.2.2. Cumulative Differences Histogram (C-DIF)

This algorithm is an improvement to the previous algorithm. The distinction is that the calculation of all differences at the beginning is not necessary [6].

The algorithm starts with the calculation of the first order differences. Then the histogram of the first differences is formed. A hypothesis test similar to the one in all differences algorithm is performed. For the detection of an emitter, two peaks are required. The requirement of the second peak at double the PRI is for improving the reliability of the detection. If no emitter is detected (in fact this will be the case for using only the first order differences), the second order differences are calculated. Then the histogram of the first and the second order differences is obtained, and the algorithm continues so on. An example of cumulative differences histogram is shown in **Figure 2.5**.

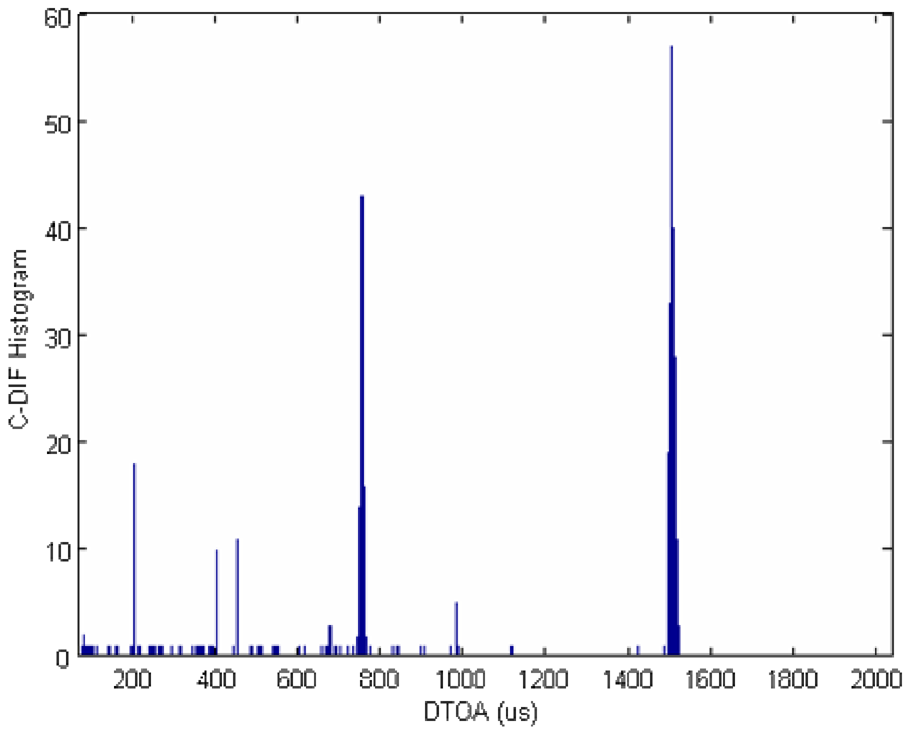

Figure 2.5 Example C-DIF Histogram

If an emitter is detected then the pulses from this emitter are removed, and the algorithm is restarted from the first order differences. The algorithm can be limited to calculate differences up to a predetermined order. If this order is reached or the number of pulses left is less than a predetermined value, the algorithm stops. The flowchart of this algorithm is given in **Figure 2.6**.

2.2.3. Sequential Differences Histogram (S-DIF)

An improvement to the cumulative differences histogram algorithm is the sequential differences histogram algorithm [7]. The distinction is that only one level histogram is used at a time.

First of all the first order differences are calculated and a histogram is formed. Then a hypothesis test is performed. The threshold used in this algorithm is different from the previous ones. In the previous algorithm threshold used was proportional to $1/PRI$, whereas in [7] it is proved that the optimum threshold for the sequential differences histogram will be proportional to e^{-PRI}. The second peak required in the cumulative differences algorithm is not used in this method, because only one level difference is available. An example of sequential differences histogram is shown in **Figure 2.7**.

If no emitter is detected by the first order differences, the second order differences are calculated. Then the histogram of the second order differences is used for the hypothesis test. The algorithm can be limited to use differences up to a predetermined order. If this order is reached or the number of pulses left is less than a predetermined threshold value, the algorithm stops. The flowchart of this algorithm is given in **Figure 2.8**.

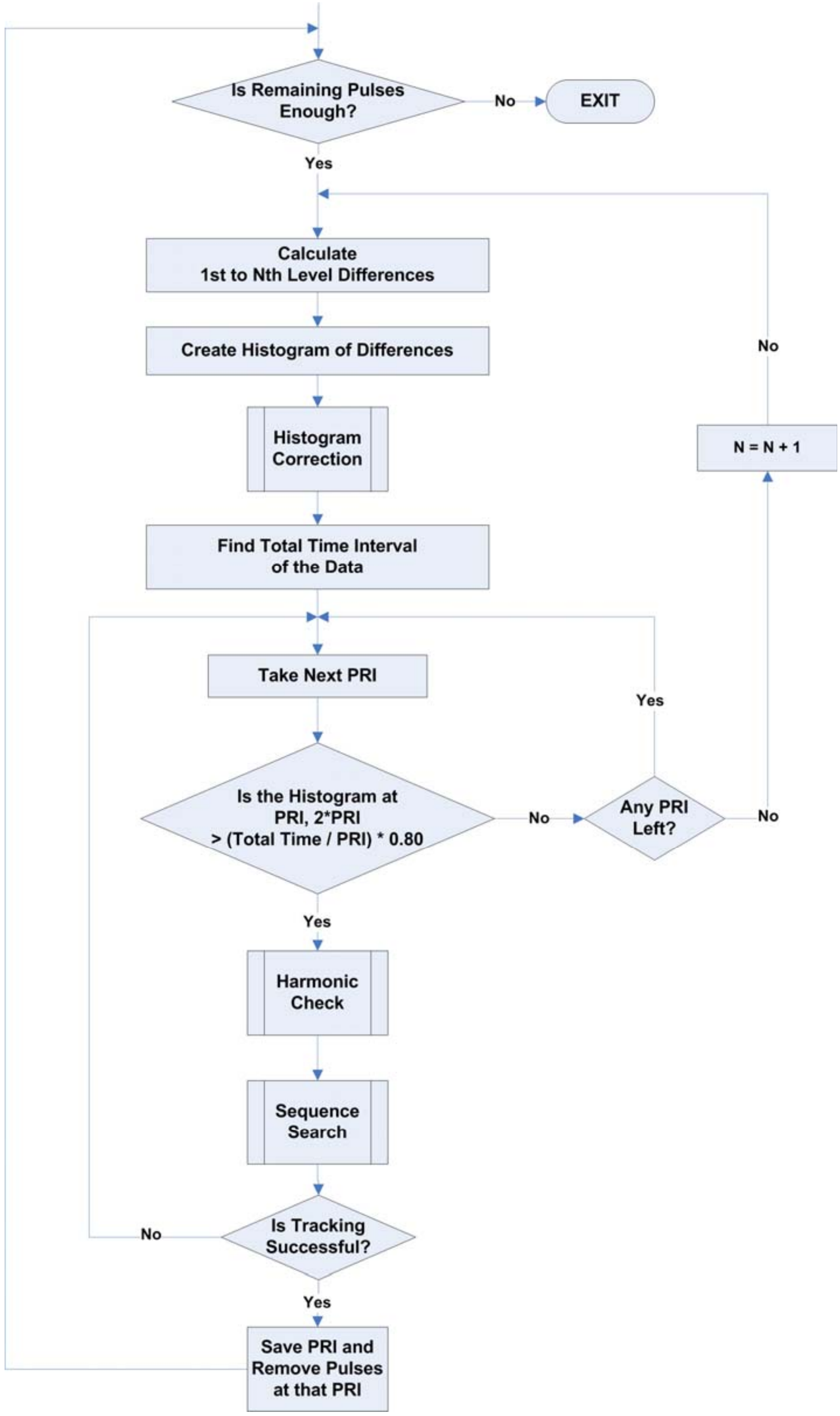

Figure 2.6 Cumulative Differences Histogram Deinterleaving Algorithm

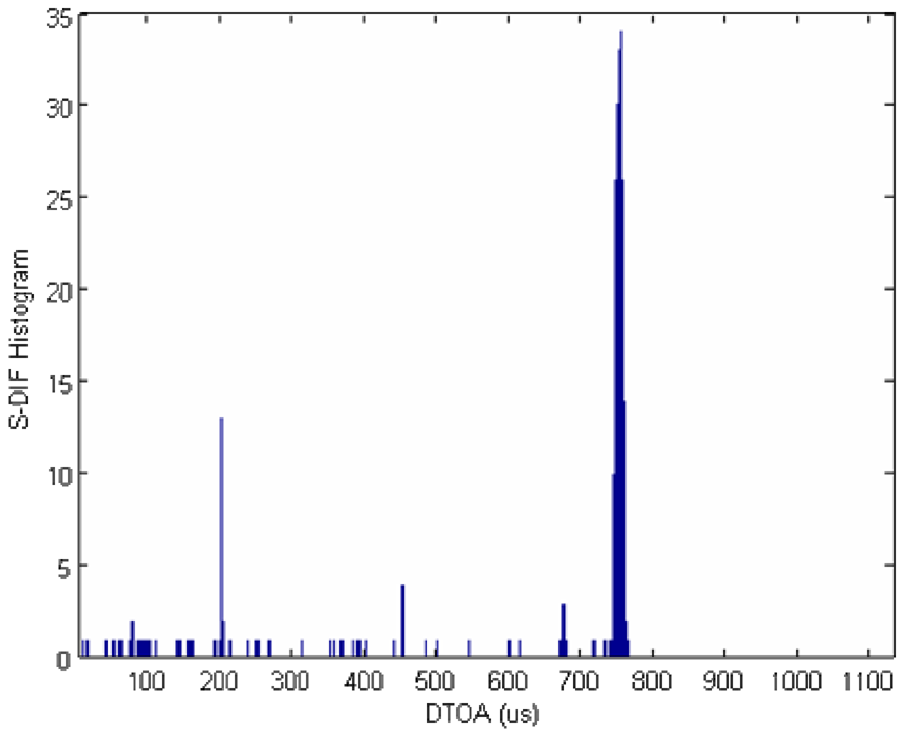

Figure 2.7 Example S-DIF Histogram

2.2.4. Improvements to the Algorithms

It is obvious that the results of these algorithms are highly dependent on the bin size of the histogram. When a small bin width is used the data may be divided into more than one bin. This violates the decision criteria, and an emitter may not be detected. On the other hand, using a larger bin width decreases the accuracy of PRI, which may cause the sequence search algorithm to fail.

A method is proposed to minimize the effect of the bin size by making a modification to the histogram. Initially, the histogram is obtained at a higher bin

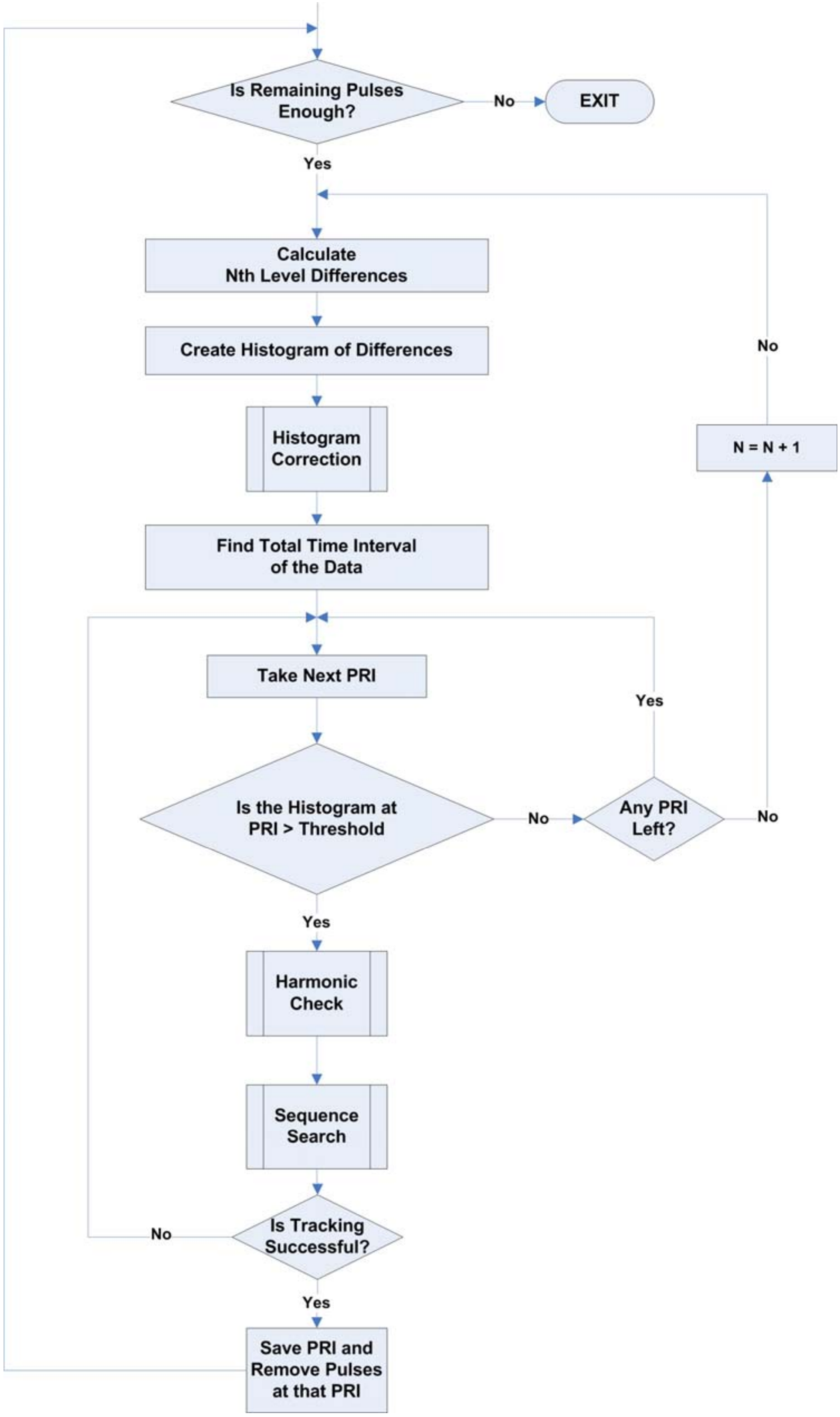

Figure 2.8 Sequential Differences Histogram Deinterleaving Algorithm

count. The peaks of this histogram are found, and the other bins around these peaks are merged. This is called a “histogram correction”. This correction will not resolve only the bin width problem but also will be useful for jittered emitters. The flowchart of the histogram correction algorithm is given in **Figure 2.9**.

In these techniques it is possible that an algorithm detects a PRI which is an integer multiple of the true PRI. This is called a “harmonic”. A simple harmonic elimination test may be based on finding the maximum of the histogram. The detected PRI is compared to the histogram maximum. If the detected PRI is at an integer multiple of the maximum, then the maximum is assumed to be the true PRI. This is the simplest example of the harmonic checking methods; there are also more complex ones.

Another powerful harmonic checking algorithm can be implemented by checking the phase of each pulse with respect to PRI. If the determined PRI is not a harmonic, all phases will be equal. The method uses the complex addition of modulus of TOAs to PRI. The following equation shows this implementation.

$$d(k) = \left| \sum_j e^{-2\pi j \frac{\mathrm{mod}(t_j, PRI/k)}{PRI/k}} \right| \quad \text{for } k = 1 \text{ to } 16 \qquad \textbf{(2.5)}$$

The argument maximizing the expression is determined. If this argument is different from unity, it means that this PRI is harmonic. The true PRI will be PRI / k_{max}.

If the PRI is not found accurately, the harmonic check may not work properly. For this reason if the maximum value is less than its expected value, the result of the harmonic check is ignored.

The flowchart of this harmonic check algorithm is given in **Figure 2.10**.

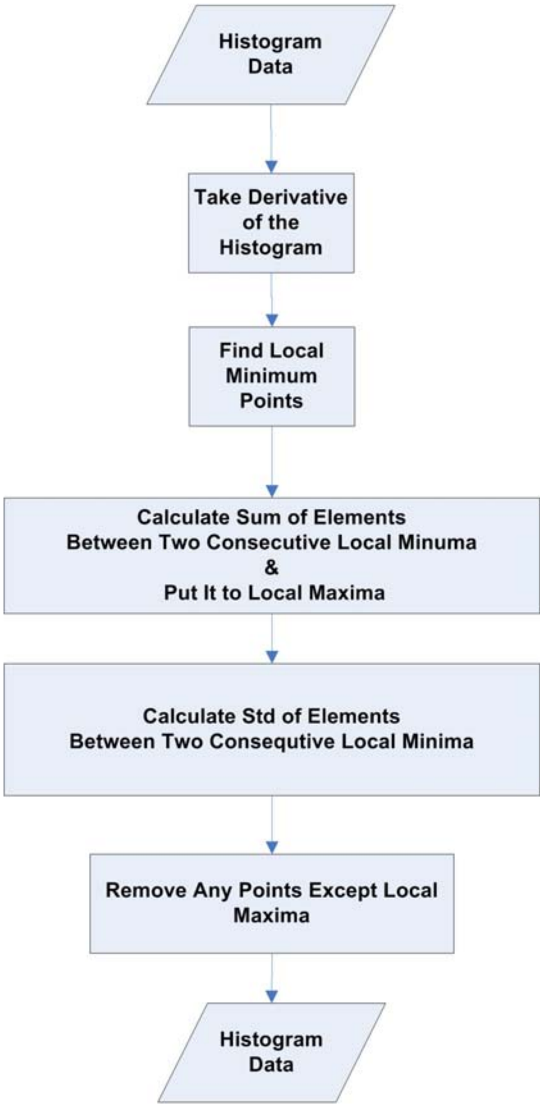

Figure 2.9 Histogram Correction Algorithm

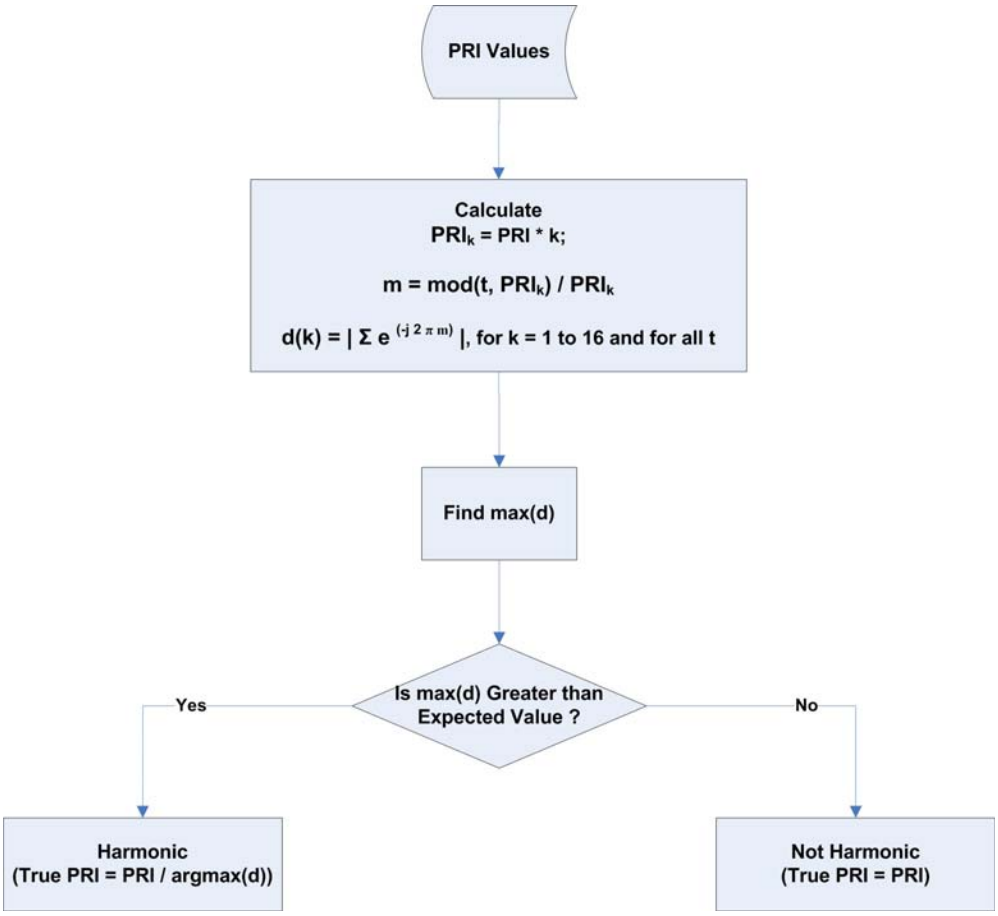

Figure 2.10 Harmonic Check Algorithm

2.3. SEQUENCE SEARCH METHOD

Another method for deinterleaving is a "sequence search" method [6]. The sequence search algorithm assumes an initial PRI estimate. The algorithm starts from the first pulse in the buffer (at t_1), and then looks for another pulse at (t_1 + PRI). If a pulse is not found, the algorithm restarts with the second pulse. If a pulse is found, the algorithm continues with this pulse and looks for the next pulse. After finding the third pulse, the algorithm continues with looking for following pulses by allowing one missing pulse. This can be implemented by looking for a pulse at (t_{last} + $2*PRI$), when the next pulse is not found. Even if the second pulse is missing, the algorithm stops and returns with the number of pulses successfully found. If the number of pulses is greater than a predefined value, then this PRI is considered as correct. If no pulses with this PRI are found, the algorithm is restarted for another PRI.

The number of consecutive pulses to start searching and the number of pulses to stop searching may be variable and algorithm may be optimized for this parameters.

This algorithm does not propose any method to determine the PRIs to be tested. The PRIs to be tested should be selected intuitively, or the PRIs to be checked may be pre-determined. It is not difficult to guess that this method is not much effective in this form. But once the PRI is known, this method can retrieve the correct pulses very simply. The major feature of the sequence search algorithm is that it presents a method for removing pulses from a block of data.

Searching for pulses at a PRI can improve the detection and decision of the other algorithms. When this algorithm is used with the histogram methods, it will bring an improvement to the decision capability of these algorithms. Also this method presents a method for removing pulses at a known PRI.

An improvement to this algorithm may be the calculation of the mean and standard deviation of all accepted pulses whenever a new pulse is accepted and then regularly updating tolerances to accept pulses in accordance with the mean and standard deviation calculated. This operation gives a precise PRI calculation. Also this method improves search of pulses with cumulative jittered PRI.

Another improvement to this algorithm may be searching in both TOA and PA. The flowchart of this version of the algorithm is given in **Figure 2.11**.

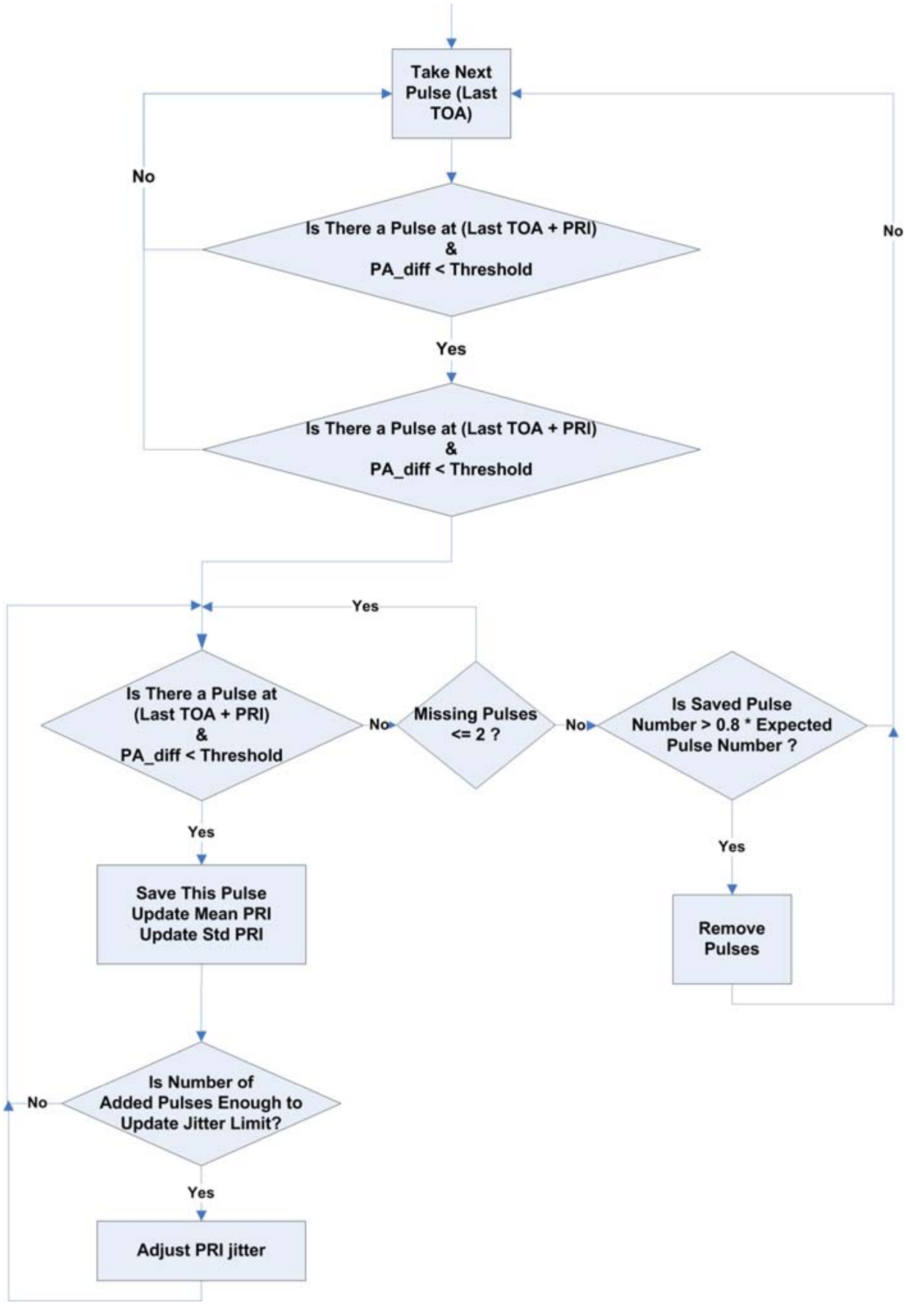

Figure 2.11 Sequence Search Algorithm in TOA & PA

2.4. WAVELET DETECTOR METHOD

Another method for deinterleaving is based on wavelet theory. This method proposes a continuous wavelet transform for detecting radar pulse sequences [9]. It operates directly on TOAs of interleaved pulse streams.

The method deals with the problem by first representing TOAs of the received signal as a superposition of continuous time impulses. A signal is assumed to be having no width and amplitude information that can be used for the deinterleaving task. Then the received signal can be given by

$$s(t) = \sum_{k=1}^{N} \delta(t - t_k) \quad \textbf{(2.6)}$$

A continuous time wavelet transform is applied to implement a detector function. The selected mother wavelet for this purpose is

$$\psi(t) = \frac{1}{\sqrt{M}} \cdot x\left(\frac{t}{M}\right) \cdot e^{j2\pi \cdot t} \quad for\ M \in \mathrm{Z}^{+} \quad \textbf{(2.7)}$$

where *x(t)* is a rectangular window of unit length; and *M* is a parameter related to the length of minimum pulse train that is desired to be detected. This expression satisfies the admissibility condition when *M* is a positive integer.

A detector function based on the wavelet transform of *s(t)* is,

$$D(T, t) = \frac{1}{M} \cdot \left| \int \psi^{*}\left(\frac{t' - t}{T}\right) \cdot s(t') \cdot dt' \right|^{2} \quad \textbf{(2.8)}$$

Simply the detector indicates whether there exists a signal with period *T* at time *t*. When the detector exceeds a threshold, a pulse train at period T is found out. It can be easily seen that the output is of the order of unity. Thus the threshold value used must be something like *0.5* or *0.6*. The threshold value is directly related to the number of missing pulses allowed.

For the *s(t)* of Eq. (2.6), the detector function can be simplified to

$$D(T,t) = \frac{1}{M^2} \cdot \left| \sum_j e^{-2\pi j \left(\frac{t_j - t}{T} \right)} \right|^2 \quad \text{for } \forall j, \quad t \le t_j \le t + MT \tag{2.9}$$

where t_j is between t and $t+MT$. This new form of the detector will simplify the implementation and will improve the performance. It is seen that the detector looks forward with rectangular window of length *MT* and makes a complex vector addition of time differences. If time differences are multiples of period T, then they will be located at the same direction. This will increase the peak magnitude and the detection will occur.

This algorithm is implemented to run for small time intervals and for the whole PRI range. For all instantaneous time values in the time interval under consideration, the detector output is calculated. Since the peaks at each time instant are determined, to eliminate the peaks having same value at different time instants, these peaks are merged. After this step the peaks are determined to be candidate PRIs, which are subject to a decision making algorithm.

Even for a single peak there will be many points exceeding the threshold, and hence a smart decision making algorithm is needed. A decision making algorithm is given in **Figure 2.13**. The algorithm should combine PRIs from the same peak and should separate PRIs from different peaks.

Once the PRI is determined, the sequence search algorithm is called for this PRI, and the pulses with this PRI are removed. The algorithm restarts to continue searching any other PRI. The flowchart of this algorithm is given in **Figure 2.12**.

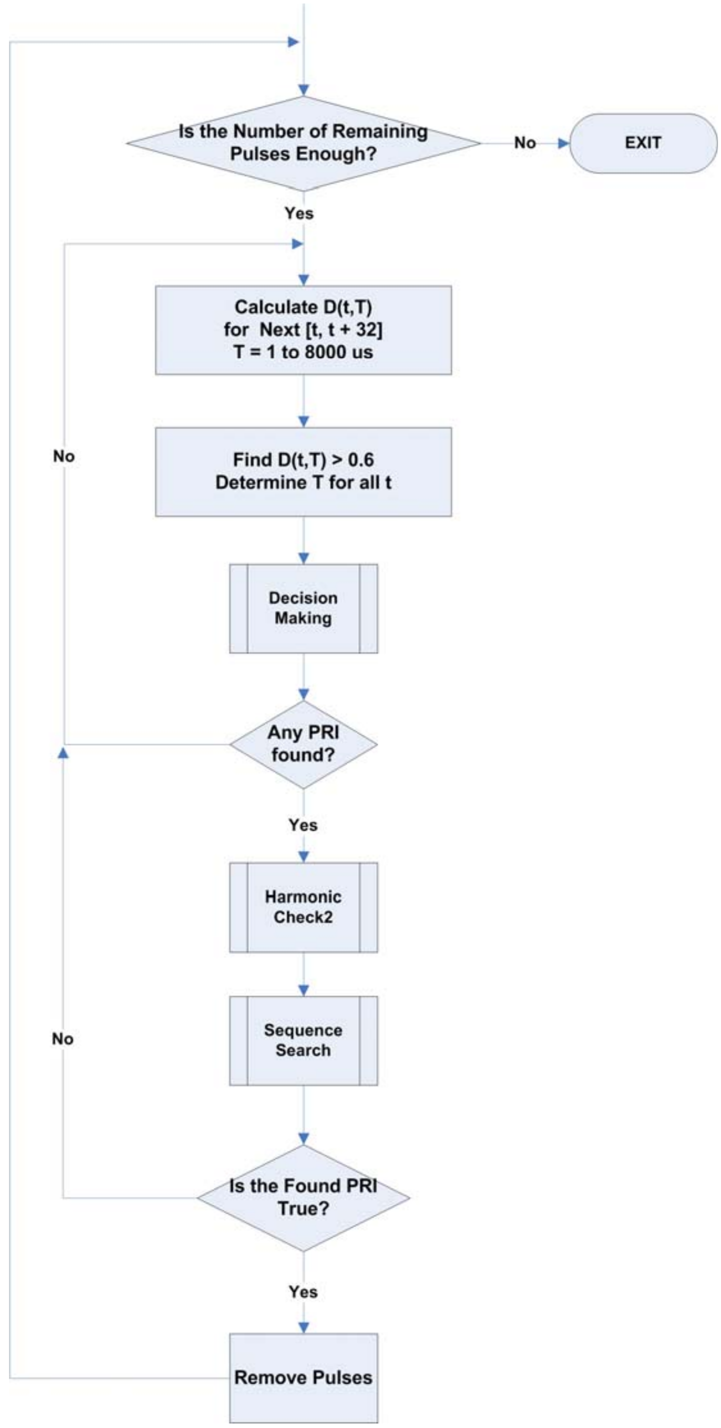

Figure 2.12 Wavelet Pulse Detector Deinterleaving Algorithm

Figure 2.13 Decision Making Algorithm of Pulse Detector

The detector output at time instant *t* is given in **Figure 2.14**. It is noticed that the peaks at 800 μs is about two times the peak at 400 μs. The peak seen at 400 μs is a harmonic of this signal (second harmonic). In this algorithm the harmonics are seen at multiples of the true PRF (not PRI). From the figure it can also be commented that the harmonic suppression of this detector is good. The third and fourth order harmonics are smaller.

Without making any modification, the wavelet method may find emitters at multiples of the true PRF. Since the sequence search could not track and retrieve an emitter with PRI less than the true value, no detection could be made. A PRF harmonic check method is proposed and used to improve the performance of the algorithm. The flowchart of this algorithm is given in **Figure 2.15**.

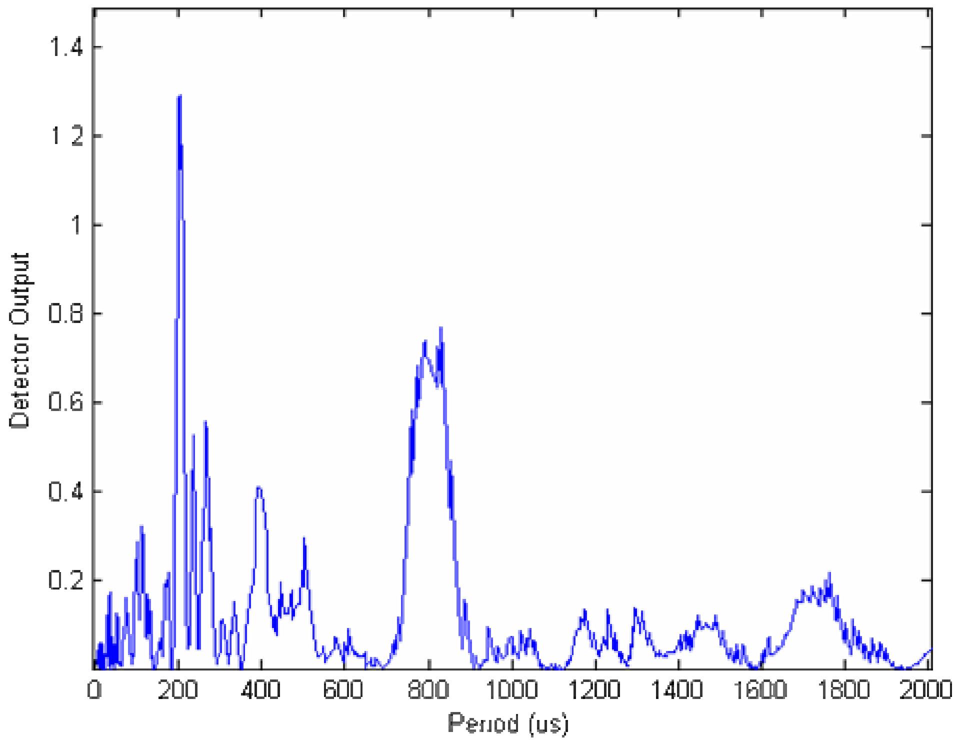

Figure 2.14 Example Pulse Detector Output at *t*

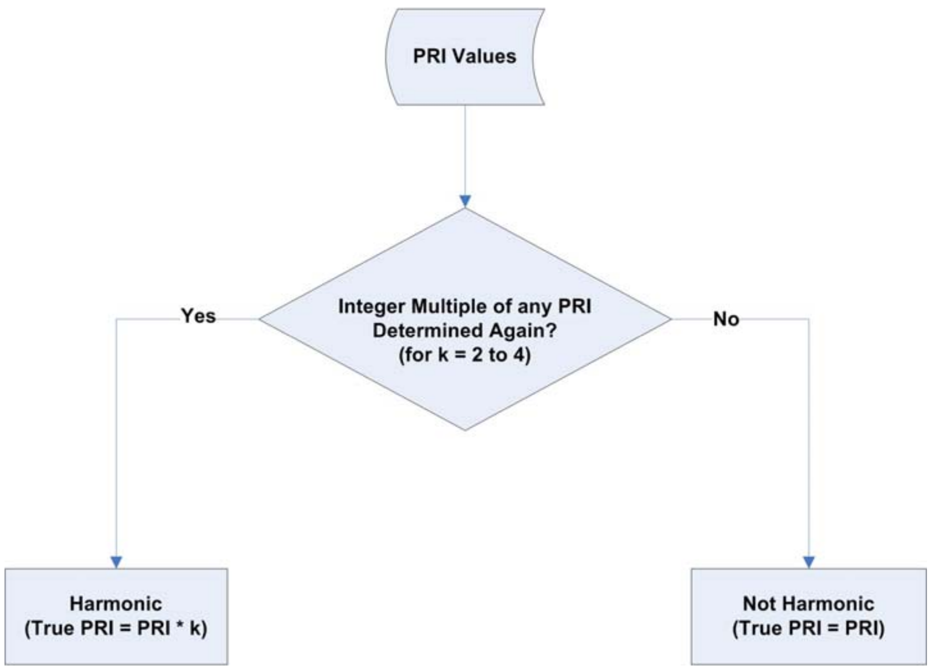

Figure 2.15 Harmonic Check Algorithm for Pulse Detector

Full output of the detector after applying the threshold is shown in **Figure 2.16**. It is seen that a PRI is changing from 190 µs to 250 µs in time. It can easily be noticed that the PRI pattern forms a rough sinusoid around a PRI of 220 µs. This shows that the detector is capable of detecting a complex sequence. Of course there are some imperfections on both signals; but this may be tolerable according to the application. Some of these imperfections can be removed using some enhancement techniques. One method to make these patterns more evident may be a change in the resolution parameter of the detector.

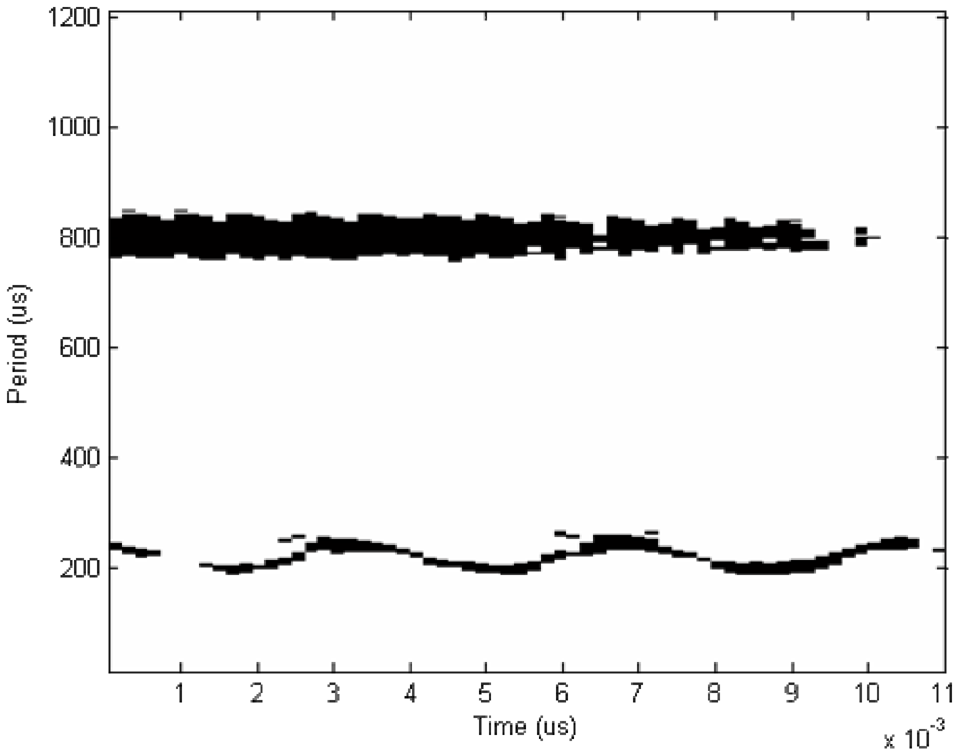

Figure 2.16 Example Pulse Detector Output

CHAPTER 3

IMPLEMENTATION OF DEINTERLEAVING ALGORITHMS

The algorithms explained in the previous chapter are implemented to see their ability for the emitter identification problem. In order to test these algorithms a simulator program is developed in MATLAB. In the following sections the designed simulator and implementations of the algorithms and their results are presented.

3.1. SIMULATOR DESIGN

In fact to have a fairly good insight about the difficulties of the deinterleaving problem, the data collected by real systems should be examined. However, since no real data is available, this is not possible for this study. Thus the environment should be simulated as close as possible to the real conditions.

The simulator is designed considering various characteristics of radar and receiver systems, which can affect the deinterleaving process. The simulator is capable of producing pulses for numerous radars in the environment. Input to the simulator is a list of radars to be simulated. The developed simulator is called a receiver simulator because it produces outputs of the receiver of EW systems, which are Pulse Descriptor Words (PDW). A PDW consists of TOA, PA, Freq, AOA and PW parameters of a pulse. The generated output is the PDWs for all of the radars to be simulated.

This simulator can simulate rotation of the radar antenna which is assumed to be rotating at a given period. The rotation period of the radar antenna affects the PA received by the simulator. The simulator can use one of 4 different antenna patterns one of which is illustrated in **Figure 3.1**. Furthermore, the pulses are assumed to be received starting from an initial random antenna position. A Gaussian noise of given variance is also added to the received PA.

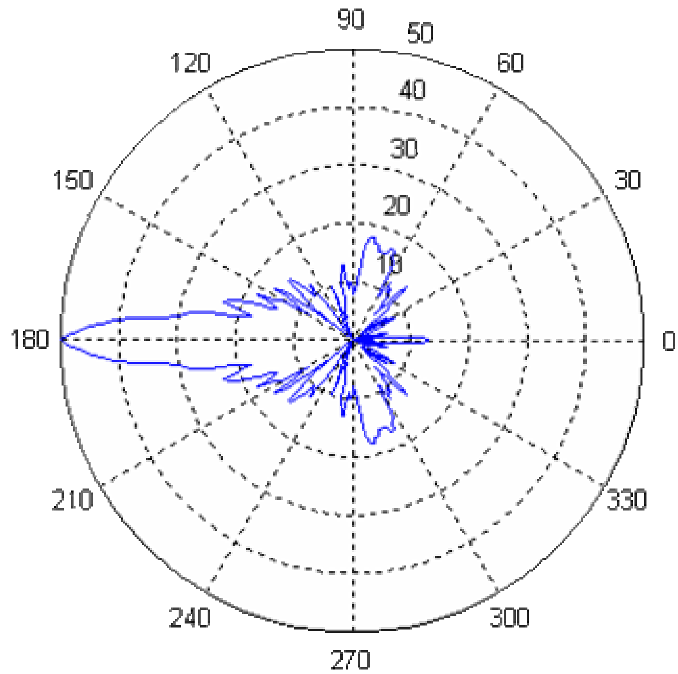

Figure 3.1 Example Antenna Pattern

The simulator supports not only regular radars but also frequency hopping and frequency agile radars. Pulse to pulse and group to group frequency changing are also supported. A Gaussian noise of given variance is added to frequency to simulate the effect of measurement noise.

The simulator supports pulse width changing radars as well. Since PW measurements can easily be affected by the PA level, a Gaussian noise is also added to the PW to simulate these measurement errors.

The simulator supports many PRI modulations such as: staggered, sinusoidal, triangular, dwell & switch, jittered and constant. A Gaussian noise of given variance is added to the TOA parameter. Simulator also supports the added jitter be cumulative.

The simulator simulates the position of the radar by assigning an AOA value to each pulse. In order to simulate the errors in direction finding stage of receiver a Gaussian noise is added to the AOA parameter.

The simulator uses the sensitivity and the dynamic range parameters of receivers to decide on whether a generated pulse can be detected. The sensitivity is the lowest power level that a receiver can measure, and the dynamic range is the power interval that the receiver can make measurements without saturation. If the PA value of a received pulse is less than the sensitivity level of the receiver, it is assumed not to be detected and ignored. Moreover, if a PA value is out of the dynamic range, it is also ignored.

The simulator generates pulses from each radar individually and then interleaves them in natural time order by combining pulses corresponding to all radars and sorting them according to their TOAs. Also the simulator takes into consideration the effect of overlapping pulses in time. It is assumed that the receiver can not receive another pulse in a time interval of [*TOA, TOA+PW*]. However, this results in missing pulses which makes the deinterleaving process difficult.

An example output of the simulator for two radars is illustrated in **Figure 3.2**. Even for this case the pulses seem to be very complicated.

The flowchart of the simulator is given in **Figure 3.3**.

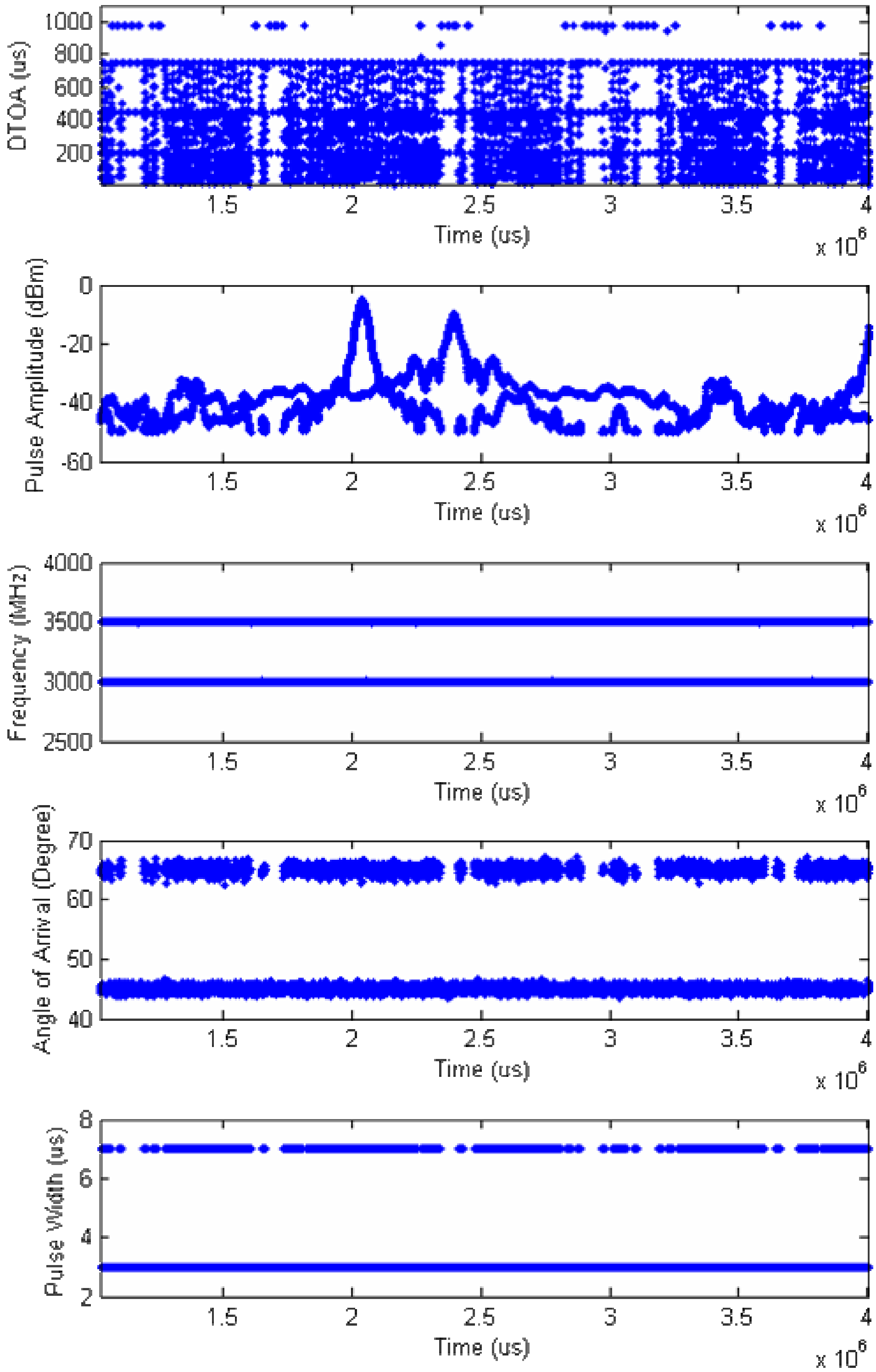

Figure 3.2 Example Simulator Output Data

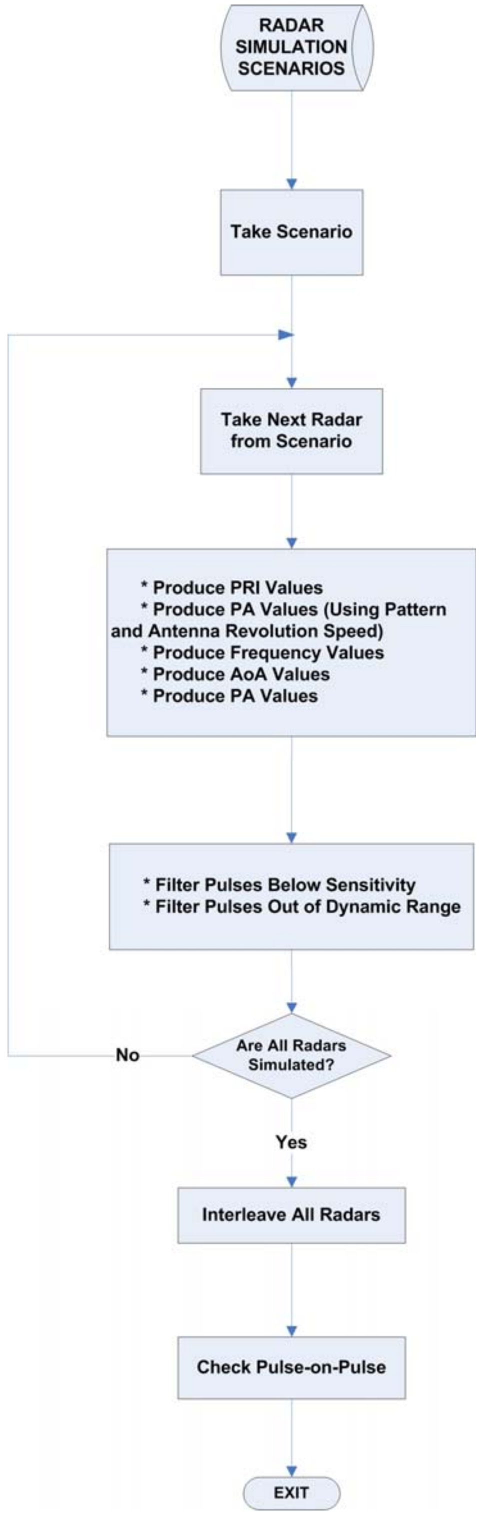

Figure 3.3 Flowchart of Receiver Simulator

3.2. EXAMPLE SCENARIOS

In this section three example scenarios are created to test and see the capabilities of the algorithms explained in the previous chapter. Using the developed simulator program, PDWs of three different example scenarios are generated. In the next section the algorithms are tested with these simulated data, and the results of each algorithm are given.

3.2.1. Example Scenario 1

The first scenario consists of four radars given in **Table 3.1**. One of the radars is a frequency hopping and another one is a frequency agile radar. One of the other two constant frequency radars is PW agile and the other is PRI agile.

3.2.2. Example Scenario 2

Another difficult scenario is introduced in **Table 3.2**. There are two radars in this scenario. One of the radars is constant frequency constant PRI radar. The other one has a sinusoidal PRI pattern and constant frequency.

3.2.3. Example Scenario 3

Two difficult radar classes to identify are dwell & switch PRI radars and staggered PRI radars. The given scenario in **Table 3.3** consists of two radars belonging to these difficult to identify classes: a 4 level dwell & switch radar and a 4 level staggered radar.

Table 3.1 Radars in Example Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4500	30	7 ± 2.8 (agile)	215	-30	2325
2	4500	50	3	300 ± 6 (jittered)	-30	1667
3	5000 / 5500 / 6000	50	7	475	-30	1052
4	4000 ± 40 (agile)	50	3	170	-30	2941

Table 3.2 Radars in Example Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3000	45	15.0	800.0	-30.0	625
2	8400	70	11.0	220.0 ± 30 (sinusoidal)	-30.0	2272

Table 3.3 Radars in Example Scenario 3

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3000	45	7	200.0 / 450.0 / 750.0 / 980.0 (dwell & switch)	-30.0	980
2	3000	60	7	1100.0 / 1115.0 / 1150.0 / 1180.0 (staggered)	-30.0	439

3.3. RESULTS OF CLUSTERING TECHNIQUES

The Improved Chain Algorithm explained in the previous chapter has been implemented using the normalized Eucledian distance. The algorithm is tested with the example scenarios. Results of the experiments are given in the following sections.

3.3.1. Results of Scenario 1

Using the Improved Chain Algorithm in its basic form (without modifications) results in **Table 3.4** are obtained. In this method the given parameters are the cluster centers. The PRI parameter given in the table is the mean DTOA of all pulses in the cluster, after removing the extreme DTOAs. It is observed that the clustering method cannot give the correct PRI value when the PRI is changing as in the cases of staggered, dwell and switch or sinusoidal PRI.

Also by this technique each frequency hop of the frequency hopping emitter is clustered separately which is an acceptable case for frequency hopping radars. On the other hand it is seen that the frequency or PW agile emitters are divided into many clusters.

Table 3.4 Results of Clustering Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (µs)**	**Pulse Repetition Interval (µs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4500.0	30.0	6.9	430.0	-29.2	1041
2	4999.8	50.0	7.0	1425.0	-29.2	328
3	5500.0	50.0	7.0	1425.0	-29.1	338
4	5999.7	50.0	7.0	1425.0	-29.2	335
5	4004.1	50.0	3.0	340.0	-29.2	1222
6	4500.0	50.0	3.0	300.0	-29.1	1576
7	3983.2	50.0	3.0	510.0	-29.1	803
8	4027.1	50.0	3.0	676.4	-29.1	509
9	3963.2	50.0	3.0	1895.4	-29.3	196
10	4047.1	49.9	3.0	5512.7	-29.3	31
11	4500.0	30.0	5.2	645.0	-29.1	545
12	4499.9	30.0	8.6	645.0	-29.1	649

After applying the modifications proposed in the previous chapter, the algorithm is run again. The results obtained are given in **Table 3.5**. It is observed that with these modifications the agile emitters are identified successfully.

Table 3.5 Results of Modified Clustering Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4009.8	50.0	3.0	340.0	-29.1	1130
2	4500.0	30.0	7.0	215.0	-29.1	2235
3	4500.0	50.0	3.0	300.0	-29.1	1576
4	4999.8	50.0	7.0	1425.0	-29.2	328
5	5500.0	50.0	7.0	1425.0	-29.1	338
6	5999.7	50.0	7.0	1425.0	-29.2	335

3.3.2. Results of Scenario 2

The Improved Chain Algorithm with modifications is applied to the second scenario. It is obvious that this scenario is very trivial for the clustering algorithm. Both of the radars are identified successfully. The results are given in **Table 3.6**.

Table 3.6 Results of Modified Clustering Algorithm for Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	8400.1	70.0	11.0	219.8	-28.9	2213
2	3000.0	45.0	15.0	800.0	-29.2	615

3.3.3. Results of Scenario 3

This scenario is also very easy for clustering algorithms. Both of the radars are identified successfully as seen in **Table 3.7**.

Table 3.7 Results of Modified Clustering Algorithm for Scenario 3

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (μs)	Pulse Repetition Interval (μs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	3000.0	45.0	7.0	449.6	-29.1	923
2	3000.0	60.0	7.0	1137.5	-29.1	428

3.4. RESULTS OF HISTOGRAM BASED TECHNIQUES

Three histogram based deinterleaving algorithms explained in the previous chapter are implemented and tested using the example scenarios.

3.4.1. Results of Scenario 1

Using these deinterleaving algorithms in their basic forms (without modifications) the results given in **Table 3.8**, **Table 3.9** and **Table 3.10** are obtained for A-DIF, C-DIF and S-DIF algorithms, respectively.

Table 3.8 Results of A-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4025.7	49.6	3.1	170.0	-29.0	2869
2	4505.6	30.3	7.0	215.0	-29.1	2224
3	4553.8	49.9	3.2	1198.7	-28.9	382
4	5226.8	50.0	5.9	1437.3	-29.2	340
5	4788.1	50.0	4.2	1832.6	-29.0	269
6	4732.0	50.0	4.0	1787.4	-29.2	280
7	4864.0	50.0	4.4	2409.5	-29.0	206
8	4868.5	50.0	4.4	3616.3	-29.0	137
9	5059.2	50.1	5.1	3684.7	-29.3	135
10	4986.2	50.0	5.0	4534.6	-29.3	109

Table 3.9 Results of C-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (μs)	Pulse Repetition Interval (μs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	4025.7	49.6	3.1	170.0	-29.0	2869
2	4505.6	30.3	7.0	215.0	-29.1	2224
3	4511.8	50.0	3.1	899.9	-28.9	513
4	5502.1	50.0	7.0	950.0	-29.0	480
5	4519.9	50.0	3.1	1799.9	-29.2	251
6	5448.8	50.0	6.8	1900.0	-29.4	242
7	4500.3	50.0	3.0	3599.7	-29.2	128
8	4515.6	50.0	3.1	3599.7	-29.3	128
9	4527.8	50.0	3.1	3599.7	-29.2	126

Table 3.10 Results of S-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4047.5	49.3	3.2	340.0	-29.2	1468
2	4508.0	30.5	7.0	430.0	-29.1	1130
3	4037.9	49.7	3.1	680.0	-29.0	718
4	4512.4	30.6	6.9	860.0	-29.2	556
5	4507.7	49.9	3.1	899.9	-28.9	526
6	4574.4	49.4	3.5	901.6	-29.1	539
7	4679.2	48.2	4.2	905.6	-29.2	531
8	4076.7	49.4	3.3	1360.0	-29.2	340
9	4977.8	45.3	6.1	1848.0	-29.1	265
10	4513.9	46.4	4.8	2705.5	-29.4	184
11	4852.9	41.0	6.5	5155.0	-29.3	95
12	4793.5	42.2	6.0	5128.9	-29.2	97

Referring to the tables, it is observed that these methods could not detect some emitters, especially the one with jittered PRI, because of the improper histogram bin size.

After applying histogram correction and harmonic check modifications explained in the previous chapter, the algorithms are run again. The results are given in **Table 3.11**, **Table 3.12** and **Table 3.13**.

In these methods PRI results given in the tables are the resultant PRI detected by the algorithms. Recall that in clustering algorithms the given PRIs are mean PRIs. Mean frequency, mean AOA, mean PW and maximum PA parameters are calculated from the extracted pulses and reported as the parameters of identified emitter. Frequency, AOA and PW parameters obtained in these methods may not be accurate as the ones in clustering algorithm; because the accidentally selected pulses by sequence search method will drift the mean of parameters away from the exact values.

Table 3.11 Results of Modified A-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4025.4	49.6	3.1	170.0	-29.0	2868
2	4505.6	30.3	7.0	215.0	-29.1	2225
3	4513.8	50.0	3.1	300.0	-28.9	1536
4	5504.9	50.0	7.0	475.0	-29.0	934

Table 3.12 Results of Modified C-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4028.4	49.7	3.1	170.0	-29.0	2869
2	4506.6	30.3	7.1	215.1	-29.1	2226
3	4512.3	50.0	3.1	300.2	-28.9	1530
4	5506.4	50.1	7.0	475.0	-29.0	934

Table 3.13 Results of Modified S-DIF Histogram Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4025.7	49.6	3.1	171.1	-29.0	2869
2	4504.7	30.3	7.0	214.9	-29.1	2222
3	4512.8	50.1	3.1	300.3	-28.9	1537
4	5504.9	50.0	7.0	474.9	-29.0	931

3.4.2. Results of Scenario 2

The histogram based algorithms with proposed modifications are applied to the second example scenario. The results of these algorithms are given in **Table 3.14**, **Table 3.15** and **Table 3.16**, respectively, for A-DIF, C-DIF and S-DIF algorithms.

It is observed that the histogram based algorithms can not detect the emitter with sinusoidal PRI pattern, because the DTOA is not stationary even for pulses of a single emitter. These techniques do not present a method for PRIs changing in time. The method given in the following section provides a means for detection of PRIs changing in time.

Also presence of such complex emitters degrades the overall performance of the algorithms. If a simple signal is incoming together with a complex signal (sinusoidal or others) the performance of the algorithm to detect the simple signal may also be degraded.

The A-DIF algorithm searches the histogram starting from small PRIs. When the peaks at 800, 1600, 3200... are detected, an emitter is found. Pulses belonging to this emitter are removed and the histogram is reformed.

For this scenario, from the created C-DIF histogram of up to third order differences, no detection can be made, because of the requirement of the peak at twice the PRI. When the C-DIF histogram of first to fourth order was formed, the algorithm detected and eliminated the emitter with PRI of 800 μs and continued processing by reforming the histogram of first order differences.

When the fourth level S-DIF histogram is formed the emitter about PRI 800 μs is detected. After removing pulses belonging to this emitter, algorithm continues with reforming the histogram of the remaining data.

In the algorithms when histogram of the remaining data is reformed, peaks of this histogram are searched. Since the DTOA is not stationary, the pulses from the sinusoidal radar will not fall into a single histogram bin, and the true PRI will not be detected. Since the threshold value used is inversely proportional to the PRI, some other bins containing higher differences will pass the threshold test and a false PRI will be detected. After removing the pulses with this false PRI, the remaining pulses will create wrong patterns which will also cause some other false PRIs to be detected.

To sum up, the most important drawback of histogram algorithms is that they can not detect an emitter which changes its PRI in time. Results are showing that all of the detected PRIs are false. Sequence search will tend to fail for all these PRIs.

Table 3.14 Results of Modified A-DIF Histogram Algorithm for Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3086.3	45.4	14.9	800.0	-29.2	625
2	8400.0	70.0	11.0	2111.7	-29.2	218
3	8399.7	70.1	11.0	2111.5	-29.3	237
4	8399.7	70.0	11.0	2816.0	-29.3	175
5	8400.4	70.0	11.0	2838.9	-28.9	161
6	8400.3	70.0	11.0	3168.1	-29.1	157
7	8400.3	70.0	11.0	3154.3	-29.1	159
8	8399.6	70.1	11.0	3178.7	-29.2	157
9	8399.7	70.0	11.0	3178.8	-29.1	156
10	8401.3	70.0	11.0	3099.0	-29.4	157
11	8399.6	70.0	11.0	3527.6	-29.2	139

Table 3.15 Results of Modified C-DIF Histogram Algorithm for Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3086.3	45.4	14.9	800.0	-29.2	625
2	8400.4	70.0	11.0	1648.2	-29.1	303
3	8400.1	70.0	11.0	2145.2	-29.3	233
4	8400.2	70.0	11.0	2485.0	-29.4	200
5	8399.8	70.1	11.0	3298.6	-29.1	151
6	8399.9	70.0	11.0	3326.4	-29.1	145
7	8399.7	70.0	11.0	3309.1	-28.9	121
8	8399.9	70.1	11.0	3300.8	-29.1	144
9	8400.6	69.9	11.0	3520.0	-29.5	66
10	8399.7	70.1	11.0	3520.1	-29.5	45
11	8400.5	70.2	11.0	3520.0	-29.2	24
12	8399.9	70.1	11.0	4945.8	-29.4	98
13	8400.4	70.0	11.0	6587.5	-29.5	67
14	8400.2	69.9	11.0	7040.1	-29.6	20

Table 3.16 Results of Modified S-DIF Histogram Algorithm for Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3086.3	45.4	14.9	800.0	-29.2	625
2	8400.2	70.0	11.0	1173.9	-28.9	422
3	8400.2	70.0	11.0	1173.8	-29.2	409
4	8399.9	70.0	11.0	1174.2	-29.1	397
5	8399.7	70.0	11.0	2346.0	-29.2	213
6	8399.7	70.0	11.0	3520.0	-29.4	137
7	8400.4	70.0	11.0	3520.0	-29.2	137
8	8400.4	70.0	11.0	3520.0	-29.1	134
9	8400.3	70.1	11.0	3520.0	-29.2	106
10	8400.0	70.1	11.0	4693.1	-29.4	107

3.4.3. Results of Scenario 3

The histogram based deinterleaving algorithms with proposed modifications are applied to the third example scenario. The results of these algorithms are given in **Table 3.17**, **Table 3.18** and **Table 3.19** respectively, for A-DIF, C-DIF and S-DIF algorithms.

The test results showed that these algorithms can not detect dwell & switch PRIs. Number of pulses in a histogram bin will be less than the number of expected pulses because the emitter does not use the same PRI every time.

For the staggered emitter, algorithms theoretically detect the frame PRI, which is the sum of the individual PRIs. Using the frame PRI, individual PRIs can be calculated via modulus operation. In this example the algorithms could not detect the frame PRI. The results found are mixture of the individual PRIs, probably because of interference of other emitter.

If in the environment there are many emitters or emitters with complex PRIs, the differences will produce a noisy background. The histogram methods can not solve this problem.

Table 3.17 Results of Modified A-DIF Histogram Algorithm for Scenario 3

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	3000.1	48.7	7.0	2796.5	-29.2	179
2	3000.1	48.9	7.0	2977.1	-29.3	168
3	2999.7	51.6	7.0	3533.5	-29.4	139
4	3000.1	50.1	7.0	4775.6	-29.1	103
5	3000.0	51.3	7.0	4494.7	-29.2	95
6	2999.9	55.2	7.0	4556.2	-29.4	78
7	3000.1	50.1	7.0	4538.8	-29.2	68

Table 3.18 Results of Modified C-DIF Histogram Algorithm for Scenario 3

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (µs)	Pulse Repetition Interval (µs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	3000.0	56.9	7.0	2274.7	-29.1	220
2	2999.9	59.7	7.0	2275.0	-29.1	220
3	3000.0	45.8	7.0	5338.9	-29.4	94
4	3000.1	45.6	7.0	7025.3	-29.2	71
5	3000.4	45.5	7.0	8138.5	-29.1	61
6	3000.0	46.5	7.0	8014.6	-29.3	60

Table 3.19 Result of Modified S-DIF Histogram Algorithm for Scenario 3

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (µs)	Pulse Repetition Interval (µs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	3000.0	48.3	7.0	1839.4	-29.1	272
2	3000.0	54.1	7.0	2273.0	-29.3	216
3	2999.9	59.0	7.0	2275.0	-29.1	198

3.5. RESULTS OF WAVELET DETECTOR

The wavelet detector explained in the previous chapter is implemented and tested with the scenarios. Obtained results for each scenario are given in the following sections.

3.5.1. Results of Scenario 1

Running this algorithm for the first scenario given in **Table 3.1**, the results in **Table 3.20** are obtained. It is seen that the algorithm can detect all PRIs also found by the histogram algorithms.

The detector output is calculated for PRIs from 1 μs to 8000 μs. The PRI searching resolution is decreased when PRI increases. Detector parameter M controls the resolution of the detector. *M* is selected to be 8 for PRIs 1 μs to 1000 μs, 12 for PRIs 1000 μs to 4000 μs and 16 for PRIs 4000 μs to 8000 μs. This selection of parameter M increases the PRI resolution. Variation in M is used to reduce the effect introduced by the decrease made in PRI search resolution, when PRI increases. The threshold used in this algorithm is 0.6.

Table 3.20 Results of Wavelet Detector Algorithm for Scenario 1

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	4025.7	49.6	3.1	170.0	-29.0	2869
2	4505.6	30.3	7.0	215.0	-29.1	2224
3	4513.8	50.0	3.1	300.0	-28.9	1535
4	5504.9	50.0	7.0	475.0	-29.0	934

3.5.2. Results of Scenario 2

For the second example scenario, the results obtained by this algorithm are given in **Table 3.21**. It is seen that this method can detect emitters when PRIs are varying in time.

In this scenario a simple and a complex sequences are interleaved. This example is important to understand the capability of the detector. It is seen that the algorithm can detect both of the signals successfully.

Table 3.21 Results of Wavelet Detector Algorithm for Scenario 2

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	8000.8	68.2	11.3	220.2	-28.9	2218
2	3930.1	49.3	14.3	800.0	-29.1	540

3.5.3. Results of Scenario 3

For the third example scenario, the results of this algorithm are given in **Table 3.22**.

It is observed that the algorithm can detect the dwell & switch PRI. Each PRI used by this type of emitter is found separately. However, these PRIs can easily be associated with each other using AOA, frequency and PW information.

This algorithm can also detect the staggered PRI which is the mean of individual PRIs. Recall that in the histogram based algorithms the output was the frame PRI, which is the summation of individual PRIs. Since the individual PRIs are closer to each other, the detector assumes them to be the same PRI. In some cases considering the mean as the resultant PRI may be meaningful. In fact it will not be a big error to consider this emitter as a simple emitter with a PRI of this “mean PRI” and having a very high jitter.

When the individual PRIs of staggered emitter are far away from each other, like in group PRI emitters, the detector may give the frame PRI. This should not be surprising because also a staggered signal may be treated as just a combination of N simple signals with the same PRI, which is the frame PRI of the staggered signal; but with different starting points. This increases the importance of the phase information. Phases of these signals are individual PRIs of the staggered signal.

In both cases the individual PRIs may be found from the phase information of pulses extracted by the sequence search algorithm.

Table 3.22 Results of Wavelet Detector Algorithm for Scenario 3

	Frequency (MHz)	**Angle of Arrival (degree)**	**Pulse Width (μs)**	**Pulse Repetition Interval (μs)**	**Max. Pulse Amplitude (dBm)**	**Pulse Count**
1	2999.9	45.1	7.0	200.0	-29.1	397
2	3000.3	45.2	7.0	371.7	-29.5	12
3	3000.1	45.4	7.0	750.1	-29.3	211
4	3000.0	45.6	7.0	980.1	-29.2	163
5	2999.9	59.5	7.0	1137.5	-29.1	423

CHAPTER 4

PROPOSED SCHEMAS FOR DEINTERLEAVING

Each of the methods in the previous chapters has its own benefit. For example the clustering method can find the mean values of frequency, AOA and PW correctly. When the emitter has frequency hopping characteristics each hop is found solely. However, when the emitter has agility in frequency or PW, results of the clustering algorithms may degrade. With modifications explained in the previous chapters, the clustering method can cope with frequency or PW agile emitters successfully.

TOA based methods deal with the PRI parameter, thus these methods can find PRI values correctly. One of the drawbacks of these algorithms is that frequency, AOA and PW parameters can not be found correctly. If some pulses are accidentally retrieved from another emitter, the mean values of frequency, AOA and PW may drift to incorrect results.

Many types of radars modulate their PRIs for many reasons, such as resolving range ambiguity or improving ECCM capability. These modulations may be in various types. Some of these PRI patterns can not be detected by histogram based algorithms. The wavelet based algorithm seems to resolve some problems encountered in the histogram algorithms; but there may still be some patterns which can not be detected. The schemas given below are proposed to resolve this problem.

4.1. PROPOSED SCHEMAS

The test results have shown that the algorithms are successful in different cases. The techniques explained in previous chapters are combined to compensate the disadvantages of them. The schemas utilize the beneficial points of these techniques. The produced schemas also propose a method for data management in an ESM system. Moreover, the use of PA is inserted to the schema to improve the reliability of the techniques. The schemas constitute of the stages explained below.

4.1.1. Data Management

In EW systems incoming data is processed by the receiver and produced PDWs are sent to processors. Since data is coming in time order, processing technique should be time based. Performance of the emitter identification task increases as the number of pulses available for identification increases. In this stage first the reception of data is waited for a predetermined time. This means that data is divided into time blocks, not into pulse blocks of fixed pulse count. The waiting time is related to the maximum PRI value expected or wished to be detected. At least 20 times the maximum PRI may be suitable. 20 is the number of pulse sufficient to identify an emitter. For reliability this number may be increased but this will cause failure in detection of short pulse trains.

After managing data in time blocks, another division of data may be in frequency, AOA or PW. It is obvious that using all three of these parameters will be more useful. This can directly be done using clustering techniques.

4.1.2. Clustering

Clustering is one of the important stages of deinterleaving schemas. By making a 3D clustering, namely frequency, AOA and PW clustering, mean and standard deviations of these parameters may be found accurately.

The clustering technique given in previous sections together with the proposed modifications is claimed to be suitable for this task. One of the important points is that it is not desired to divide any of emitters into two different clusters, except for frequency hopping ones. If an emitter is divided into two or more parts it can not be detected in the next stages which can adversely affect the performance of the next sections.

The produced clusters will never be destructed and will be reused whenever new data comes. After incoming data is distributed into clusters, the number of pulses in each cluster is checked. If one of the clusters does not have sufficient number of pulses, pulses in that cluster will be ignored and will not be processed in the following stages. Other clusters will be processed successively.

4.1.3. Emitter Tracking (Pre-Tracking)

Another important stage is the tracking of known emitters. Known emitters list may be inherited from a database library, entered by an operator or from the previous deinterleaving results.

The proposed tracking technique is a multi parameter sequence search algorithm in both TOA and PA. PA is the key parameter for emitter identification. At first look PA may not be considered as a good deinterleaving parameter, because of its varying nature with the rotation of antenna. But since rotation of the antenna is very slow with respect to the pulse rate, the pulse to pulse PA change will be small (except for some electronically scanning radars). If two radars are received at the same time, their PAs mostly will not be of the same level, which allows a tracking in PA. Perhaps the received interleaved sequence will be constituted of main lobe of one radar and side or back lobe of another radar. In this case PA will be a major parameter for deinterleaving and tracking.

Tracking will be made for each cluster solely. If an emitter is known to be belonging to one of the clusters, it will be tried to be tracked within pulses from that cluster.

4.1.4. PA For Deinterleaving

PA is an important parameter not only for tracking but also for deinterleaving. Up to this point pulses collected in a cluster for a period of time have been tracked and pulses from known emitters are extracted. If the left pulses are not sufficient to detect a new emitter, no more processing will be done on these pulses. If there exist a sufficient number of remaining pulses, they will be used for detection of any unknown emitters. The problem in detection stage is finding the most appropriate data for processing. PA will be a key parameter for determining the best partition of data. In fact the best data will be a single illumination of an emitter. To determine the appropriate data, pulses below a PA level are temporarily discarded; therefore filtered pulses are not used for emitter detection purpose, but used for tracking purpose at the next stage. Time differences of pulses are calculated and the ones greater than an experimentally determined value are considered as time difference between consecutive illuminations. Data is divided into blocks from those relatively great time gaps and time gaps are removed. Then the maximum PA level of each block is calculated, and the one with maximum PA level is selected for the next stage. This is the estimation for the main lobe of an emitter. Processing in the next stages will give better results if main lobe could have been determined.

4.1.5. Deinterleaving and PRI Estimation

For PRI estimation and for deinterleaving (if interleaved emitters still exists) one of the four methods explained in previous chapters can be used. Data from previous stages will be processed in one of the algorithms and PRIs of emitters will be determined. The result and performance of each of the histogram method and wavelet detector will be discussed in the next section. For histogram algorithms histogram correction will be applied. When a PRI is determined, a harmonic check will be made. Note that in the histogram based methods harmonics are the PRI harmonics (that is integer multiple of true PRI) and in wavelet based method algorithm harmonics are the PRF harmonics (that is integer multiple of PRF, a fraction of PRI). If an emitter is detected pulses from this emitter will be extracted with sequence search in TOA. After extracting a detected emitter, the remaining data is searched for other emitters. Algorithm stops when no detection occurs. Detection may not occur because all emitters found or data remaining is more complicated to make a decision.

4.1.6. Sequence Search (Post-Tracking)

After all emitters are detected from a single block (selected with checking PA level in the previous stage) these new detected emitters will be tracked in all data. PA filter applied in the previous stage and division of data into blocks will be canceled for this stage. After all emitters are tracked, remaining data is again send to PA deinterleaving stage for any other emitters remaining in other lobes that are not processed in the deinterleaving stage.

Deinterleaving is made only for a block of pulses (which is determined in PA stage.) but tracking is made over all data in the cluster. Note that after all processing there may even be remaining pulses. All pulses associated with an emitter are signed by an emitter number. Pulses that could not be related to an emitter will be given only a cluster number. By this approach emitters with more complex PRI patterns may be identified (except for PRI).

After all data is processed, determined emitters are saved to be used in the next iteration for pre-tracking purpose. Then another cluster is taken for processing and schema above is repeated. After all clusters are processed, new data is waited again from the receiver for another time period and all processing restarts.

4.1.7. Emitter Combining

Since frequency hopping emitters can be divided into two or more clusters, its each hop may be found as a single emitter. If frequency hopping is pulse to pulse, the PRI of each emitter will be found to be hop count times the true PRI. By checking the PRI, AOA and PW, these emitters may be linked to be a single radar. If frequency hopping is group to group, the PRI will be same in all groups thus PRI may be detected correctly (only if pulse count in each group is sufficient). Again these emitters may be linked to be a single radar. If the radar is changing both its frequency and PRI at the same time, all emitters will be detected solely. In this case emitters can only be linked by their AOA and PW values. However this information may not be sufficient for linking the emitters, because there are platforms carrying multiple radars and emitters will be found as coming from same direction. PW will remain as the single parameter emitter linking and it will not be a reliable parameter.

In histogram methods and sometimes in wavelet detector method staggered PRI emitters will be found as the frame PRI. However their individual PRIs may be found simply calculating phase of the pulses. Linking of these emitters may be simply done by checking emitters found from the same cluster and at the same PRI.

Dwell & Switch emitters detected in wavelet technique may be linked in the same manner. They will be found from the same cluster and each emitter will be received just after the other emitter.

Linking of emitters is a sensitive subject. Some emitters may be accidentally linked. To prevent accidental links, emitter linking must be made using a database library or with the help of an operator using the ESM system.

In this thesis study emitters are not linked automatically. It is assumed that these schemas determine emitters (not radars) which may be the radar itself or only a mode of the radar.

4.2. IMPLEMENTATION OF SCHEMAS

According to the proposed methods in the previous section, 3 different groups of schemas for emitter identification have been implemented. The first group constitutes of one clustering based schema. This schema only divides the data into time blocks and applies the clustering algorithm. The schemas in the second group divides the data into time blocks, applies PA deinterleaving, that is illumination finding and PA filtering, and followed by one of the A-DIF, C-DIF, S-DIF histogram methods or the wavelet detector method. These schemas also use tracking in TOA and PA. The schemas in the third group first divides the data into time blocks, next apply clustering, then PA filtering and illumination finding and finally one of the A-DIF, C-DIF, S-DIF histogram methods or the wavelet detector method. These schemas also use tracking in TOA and PA. **Table 4.1** gives a list of the methods used by schemas.

Simplified flowchart of these proposed schemas are given in **Figure 4.1**, **Figure 4.2** and **Figure 4.3**

Table 4.1 Feature Table of Several Schemas

<table>
<tr><th>Schema</th><th>Clustering</th><th>Tracking</th><th>PA Processing</th><th>Deinterleaving</th></tr>
<tr><td>1</td><td>Improved Chain + Cluster Merging</td><td>X</td><td>X</td><td>X</td></tr>
<tr><td>2</td><td rowspan="4">X</td><td rowspan="8">In TOA and PA</td><td rowspan="8">PA Filtering + Illumination</td><td>A-DIF Histogram</td></tr>
<tr><td>3</td><td>C-DIF Histogram</td></tr>
<tr><td>4</td><td>S-DIF Histogram</td></tr>
<tr><td>5</td><td>Wavelet Detector</td></tr>
<tr><td>7</td><td rowspan="4">Improved Chain + Cluster Merging</td><td>A-DIF Histogram</td></tr>
<tr><td>7</td><td>C-DIF Histogram</td></tr>
<tr><td>8</td><td>S-DIF Histogram</td></tr>
<tr><td>9</td><td>Wavelet Detector</td></tr>
</table>

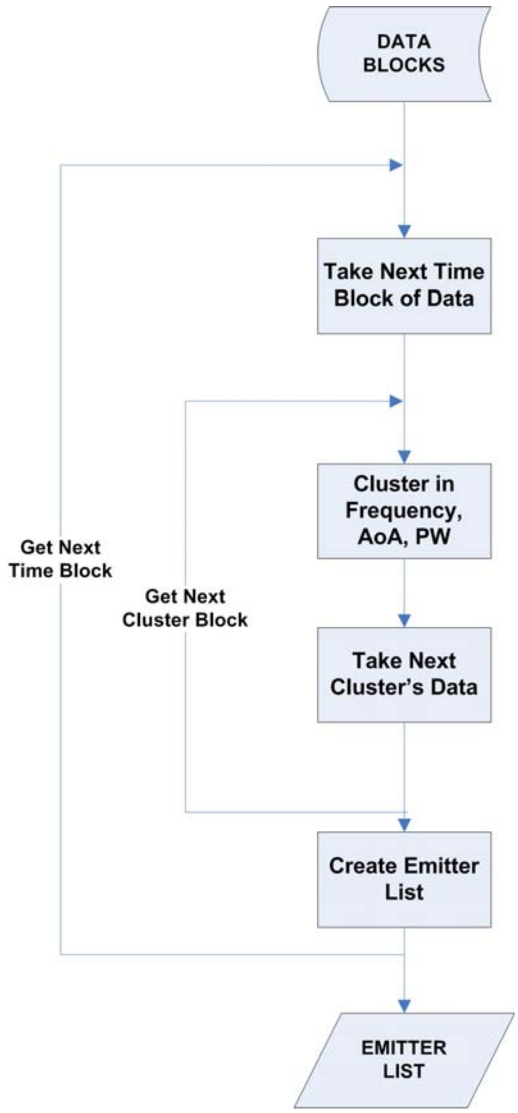

Figure 4.1 Clustering Based Schema

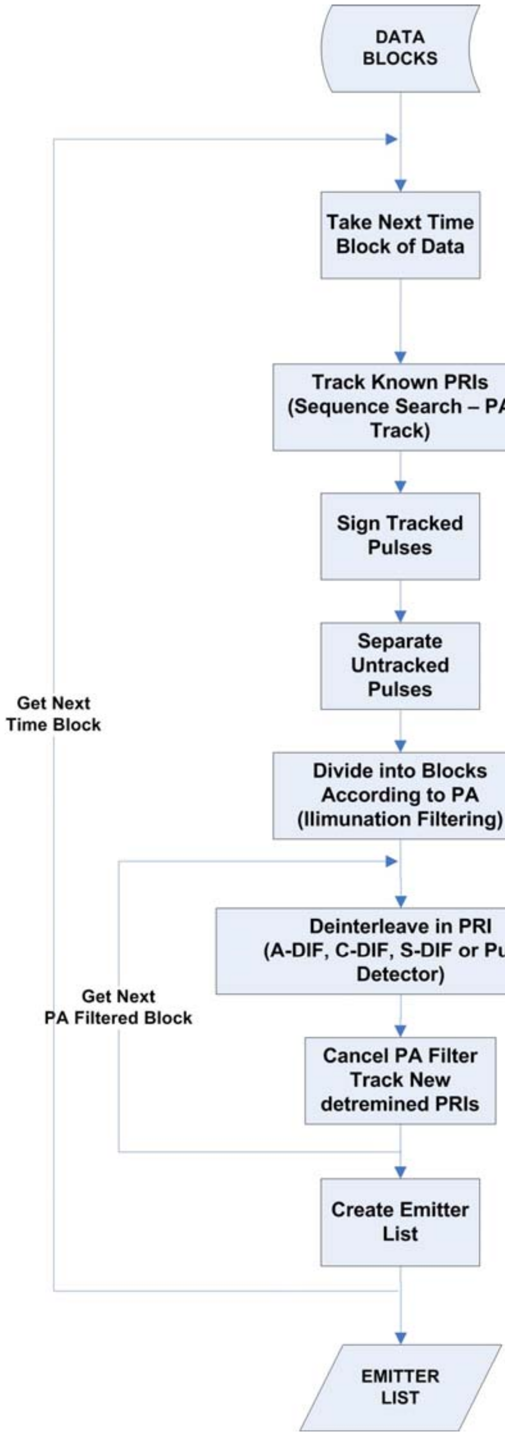

Figure 4.2 TOA Deinterleaving Based Schema (Second Group)

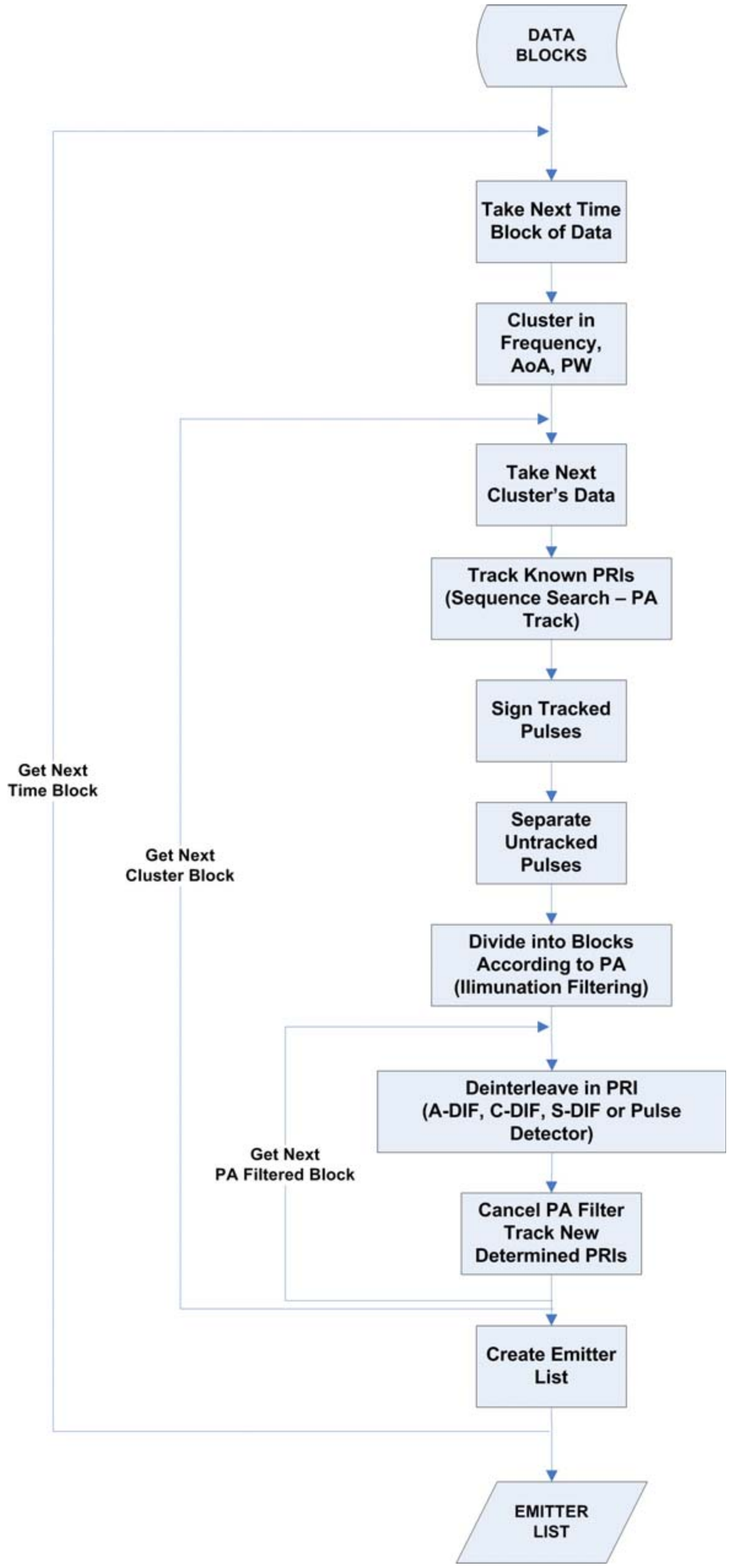

Figure 4.3 Clustering and TOA Deinterleaving Based Schema (Third Group)

4.3. TEST SCENARIOS

To test these schemas, three test scenarios are created using the simulator program described in the previous chapter. Implementations are made in MATLAB high level programming language, and the tests had been run on an Intel Pentium 4 / 2.80 GHz computer.

4.3.1. Test Scenario 1

Number of emitters in this scenario is 8. Radars are simulated for 10 seconds. Antenna patterns are randomly selected from 4 different available patterns. Other parameters of simulated radars are given in **Table 4.2**. The total number of pulses in this scenario is 22091.

4.3.2. Test Scenario 2

This scenario consists of 11 radars which are simulated for 10 seconds yielding 84850 pulses. Other parameters of simulated radars are given in **Table 4.3**.

4.3.3. Test Scenario 3

This scenario consists of the 19 radars in the previous two scenarios. These radars are simulated for time duration of 20 seconds. Notice that increasing simulation time increases the number of pulses received from each radar, hence increases the processing time for schemas. The total number of pulses in this scenario is 198227.

Table 4.2 Radars in Test Scenario 1

	Frequency (MHz)	**AOA (degree)**	**PW (μs)**	**PRI (μs)**	**Max. PA (dBm)**	**ARP (sec)**	**Pulse Count**
1	4500.0	30.0	7.0 ± 2.8 (agile)	215.0	-20.0	1.0	1765
2	4500.0	50.0	3.0	300 ± 6 (jittered)	-25.0	5.0	7655
3	5000.0 / 5500.0 / 6000.0	50.0	7.0	475.0	-20.0	4.0	1588
4	4000.0 ± 40.0 (agile)	40.0	3.0	170.0	-23.0	3.0	1887
5	3000.0	35.0	15.0	800.0	-19.0	5.0	8642
6	8400.0	20.0	11.0	220.0 ± 30 (sinusoidal)	-30.0	7.0	518
7	3000.0	45.0	7.0	200.0 / 450.0 / 750.0 / 980.0 (dwell & switch)	-30.0	8.0	256
8	3000.0	65.0	7.0	1100.0 / 1115.0 / 1150.0 / 1180.0 (staggered)	-35.0	6.0	62

Table 4.3 Radars in Test Scenario 2

	Frequency (MHz)	**AOA (degree)**	**PW (μs)**	**PRI (μs)**	**Max. PA (dBm)**	**ARP (sec)**	**Pulse Count**
1	3000.0	5.0	7.0 / 20.0	175.0	-9.0	7.0	6474
2	3000.0	45.0	15.0	800.0	-17.0	9.0	679
3	4000.0	5.0	3.0	145.0 ± 4.8 (jittered)	-20.0	3.0	46748
4	4000.0 ± 40.0 (agile)	30.0	3.0	105 ± 4.2 (jittered)	-15.0	5.0	25332
5	5100.0 ± 51.0 (agile)	55.0	11.0	100.0 / 250.0 / 550.0 / 550.0 (staggered)	-25.0	5.0	502
6	5100.0	85.0	11.0	200.0 / 450.0 / 750.0 / 980.0 (dwell & switch)	-15.0	3.0	1472
7	6800.0	70.0	80.0	440 to 580 (sliding)	-35.0	12.0	470
8	8400.0	70.0	11.0	220.0 ± 30.0 (sinusoidal)	-22.0	8.0	2677
9	6400.0 / 6600.0 / 6800.0	70.0	20.0	335.0	-31.0	10.0	415
10	7800.0 / 8400.0	85.0	15.0 / 30.0	1050.0	-29.0	10.0	477
11	9800.0	30.0	1.0	80.0	-30.0	3.0	2455

4.4. TEST RESULTS

For the first schema, emitters are assumed to be identified if a cluster includes pulses only from one emitter and if PRI is consistent. Otherwise, the identified emitter will have an incorrect PRI value.

For staggered PRI emitters, the histogram methods find the frame PRI. The wavelet detector may find either the frame or the mean PRI, according to the amount of variation in the individual PRIs. Both are accepted to be successfully identified.

A Dwell & Switch PRI emitter is accepted to be identified successfully if its all PRIs are determined.

Frequency Hopping emitters are accepted to be identified if all frequency hops are determined.

4.4.1. Test Results 1

All of the schemas have been run with the PDWs generated for Test Scenario 1. Results obtained from Schema 3 are given in **Table 4.4**.

The emitters seen at rows 1, 2, 3, 6 and 14 to 21 are successfully identified by this schema, which correspond to radars 4, 1, 3, 5 and 6 in **Table 4.2**. Emitters 14 to 21 correspond to individual PRIs of radar 6 which is found as staggered PRI. In fact the radar 6 has a sinusoidal PRI pattern but it is reported as staggered PRI. For the reason that its PRI variation period is 8 pulses, it can also be seen as 8 level staggered. Hence it is counted to be successfully identified.

The total number of successfully identified radars by this schema is 5. Since the scenario consists of 8 radars, 3 radars are missed. Other 9 emitters seen in the table are false emitters. The emitter 5 is counted to be a duplicate emitter because it is just a duplicate of emitter 4, where the emitters 4 and 5 are harmonics of radar 2 which has jittered PRI pattern.

The emitters obtained from Schema 9 are given in **Table 4.5**. It is obvious that this schema could identify all radars successfully, where the emitters 5 to 8 are corresponding to the Dwell & Switch PRI emitter and 11 to 13 are corresponding to the Frequency Hopping emitter.

The results of all schemas are summarized in **Table 4.6**. In the first schema 7 of 8 radars from scenario 1 are successfully identified and one of the radars is missed. The missed radar is the dwell & switch radar. Instead of the dwell & switch radar this schema reported a radar with a PRI value, which is the mean of all PRIs of the dwell & switch radar.

The schemas from 2 to 4 can identify some emitters successfully but can not detect some others. Notice that the number of false emitters is high. Also there are some duplicated emitters, which should also be counted as false emitters.

A duplicated emitter can be either the duplication of a successfully reported emitter or the duplication of a false emitter. In these methods jittered, sinusoidal and dwell & switch emitters are not detected and false emitters and duplicated emitters arise because of these emitters. These types of emitters sometimes can not be tracked and some leakage pulses pass to the next stage. Remaining pulses are again deinterleaved and an emitter is found which is either a false emitter or the duplicate of another emitter. Usually false emitters can not be tracked successfully, a new false emitter is detected from these untracked pulses, which causes a new false emitter.

Table 4.4 Results of Schema 3 for Test Scenario 1

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (μs)	Pulse Repetition Interval (μs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	4000.2	50.0	3.0	170.0	-22.7	1849
2	4499.0	30.0	7.0	215.0	-19.4	1545
3	5491.4	50.0	7.0	475.0	-19.6	1558
4	4496.9	49.9	3.1	600.1	-24.9	3643
5	4486.0	49.9	3.1	600.0	-24.7	3573
6	3000.2	35.0	15.0	800.0	-19.1	6453
7	5220.6	37.8	7.5	2117.3	-30.0	199
8	7573.0	26.4	9.3	2462.0	-30.1	66
9	5544.2	35.7	7.9	2209.9	-29.5	135
10	3000.4	45.0	7.0	2249.2	-30.6	20
11	6802.6	28.7	9.3	2463.9	-29.7	77
12	6747.2	25.9	11.5	3426.4	-30.2	43
13	7500.6	24.1	10.8	3566.0	-30.5	37
14	8400.0	20.0	11.0	3519.9	-29.8	18
15	8400.0	20.0	11.0	3520.0	-30.2	18
16	8399.8	20.0	11.0	3520.0	-30.3	24
17	8402.0	20.0	11.0	3520.1	-31.7	24
18	8399.4	20.0	11.0	3520.1	-32.2	21
19	8400.5	20.1	11.0	3520.1	-30.1	12
20	8400.1	20.1	11.0	3520.1	-30.5	6
21	8397.3	20.0	11.0	3520.1	-30.7	10
22	8401.7	20.0	11.0	7040.3	-30.7	5

Table 4.5 Results of Schema 9 for Test Scenario 1

	Frequency (MHz)	Angle of Arrival (degree)	Pulse Width (μs)	Pulse Repetition Interval (μs)	Max. Pulse Amplitude (dBm)	Pulse Count
1	4000.7	50.0	3.0	170.0	-22.7	1742
2	4500.0	30.0	7.0	215.0	-19.4	1518
3	8400.4	20.0	11.0	220.2	-29.7	516
4	4500.0	50.0	3.0	300.0	-24.7	7107
5	3000.0	45.0	7.0	200.0	-31.6	97
6	3000.0	45.0	7.0	450.0	-30.4	60
7	3000.0	45.0	7.0	749.9	-29.5	50
8	3000.0	45.0	7.0	979.9	-31.3	43
9	3000.0	35.0	15.0	800.0	-19.1	6456
10	3000.2	65.1	7.0	1136.8	-35.2	62
11	4999.8	50.0	7.0	1425.0	-19.8	513
12	5500.2	50.1	7.0	1425.0	-19.9	513
13	6000.2	50.0	7.0	1425.0	-19.6	516

The schema 5 with the wavelet detector gives slightly better results. Notice that the number of false emitters is low compared to the histogram based schema.

The schemas from 6 to 8 give nearly the same results with the schemas 2 to 4. The main difference is in processing time. This shows that combination of clustering into histogram based methods improves the performance of algorithms.

The last schema has detected all emitters successfully with the processing cost of about 5 times more.

Table 4.6 Results of Deinterleaving Schemas for Test Scenario 1

Schema	Number of Successfully Identified Emitters	Number of Missed Emitters	Number of False Emitters	Number of Duplicated Emitters	Processing Time (Sec.)
1*	7	1	1	0	4.2735
2	5	3	11	1	10.2775
3	5	3	9	1	13.1184
4	6	2	3	1	8.4454
5	5	3	2	0	59.4964
6	5	3	11	1	6.1908
7	5	3	11	2	9.7845
8	6	2	5	1	6.1345
9	8	0	0	0	44.7579

* Identified without PRI

4.4.2. Test Results 2

The results given in **Table 4.7** have been obtained by running all of the schemas with the PDWs generated for Test Scenario 2.

The results obtained in this experiment are similar to the results of the previous experiment. The emitters with complex PRI are either not detected or duplicated. Also it is observed that the number of false and duplicated radars is increased when the environment gets more complicated.

Table 4.7 Results of Deinterleaving Schemas for Test Scenario 2

Schema	Number of Successfully Identified Emitters	Number of Missed Emitters	Number of False Emitters	Number of Duplicated Emitters	Processing Time (Sec.)
1*	9	2	2	0	18.4458
2	5	6	31	13	70.927
3	5	6	30	11	66.5457
4	3	8	21	6	48.9183
5	6	5	17	12	330.3201
6	8	3	16	42	39.485
7	7	4	17	55	64.0593
8	6	5	16	29	44.2333
9	11	0	2	0	120.5254

* Identified without PRI

4.4.3. Test Results 3

The proposed schemas have been tested with the Test Scenario 3 which consists of 19 radars. This scenario is a very complicated scenario for EW systems. The results obtained for this scenario are given in **Table 4.8**.

The results of the first schema are similar to the results in the previous example. The main difference is in processing time.

The processing time spent for each schema is much higher than the previous ones. Furthermore it is wrong to expect processing time to be twice (since simulation time is doubled) the total processing time of the previous examples. This is expected because these algorithms are not linear. When the environment gets complicated, the processing time increases exponentially.

Table 4.8 Results of Deinterleaving Schemas for Test Scenario 3

Schema	**Number of Successfully Identified Emitters**	**Number of Missed Emitters**	**Number of False Emitters**	**Number of Duplicated Emitters**	**Processing Time (Sec.)**
1*	16	3	2	0	61.8537
2	4	15	42	8	796.4167
3	2	17	41	15	635.7099
4	4	15	24	4	383.3756
5	9	10	30	6	736.6045
6	10	9	29	77	121.711
7	9	10	33	91	193.1853
8	11	8	27	64	103.7832
9	18	1	26	9	503.3875

* Identified without PRI

Another important point is observed when the results of schemas 2, 3, 4 and 5 are compared to the results of schema 6, 7, 8 and 9. The only difference between these groups of schemas is that the first group does not utilize the clustering technique. The results of the first group of schemas are very poor even though they consume more processing time. The schemas utilizing the clustering technique give better results both in detection capability and in processing time.

When the histogram based schemas are compared to each other, the results are very similar. The wavelet detector based schema seems to give better results even in complicated environment. However this schema consumes more processing power.

These experiments show that the last schema, which is based on the wavelet detector, gives best results. It seems that combining the wavelet detector based method with a clustering technique improves the performance of this method. Other techniques proposed in this schema increase the reliability of the schema. It is observed that applying pre-tracking and post-tracking decreases the processing time of the total schema. Usage of PA for deinterleaving relieves the wavelet pulse detector.

CHAPTER 5

CONCLUSION & FUTURE WORK

In this thesis, several techniques are studied to solve the emitter identification problem in Electronic Warfare systems.

The previous approaches to this problem, namely, the clustering based method, the histogram based methods and the wavelet based method are investigated.

There are many clustering based methods known to be applied to the emitter identification problem. Only one of these methods, the Improved Chain Algorithm is selected, to examine the ability of clustering based methods. An adaptive method to update the limits of clusters and a cluster merging method is proposed to improve the results of the Improved Chain Algorithm.

DTOA histogram based methods are types of TOA based deinterleaving algorithms. There are three different versions of DTOA histogram based algorithms, which are All Differences Histogram Algorithm, Cumulative Differences Histogram Algorithm and Sequential Differences Histogram Algorithm. All of these methods use DTOA histograms to estimate PRIs of the emitters in a given interleaved sequence and thus extracting pulses from each emitter. It is observed that results of these algorithms are highly dependent on the histogram bin sizes. A method is proposed to adjust histogram bin sizes. Since these algorithms use higher order differences, the harmonic PRIs can also be detected. A method is combined with the algorithms to check whether the detected PRI is harmonic.

Another type of TOA based deinterleaving algorithm is the wavelet based pulse detector. This algorithm uses a detector that decomposes a pulse sequence into small overlapping partitions and makes use of complex addition of phases of pulses in each partition with respect to the PRI under test. When the detector output exceeds a certain threshold, PRI is determined and the emitter with that PRI can easily be extracted from the input pulse sequence. A complicated decision making method is proposed to increase the reliability of the detector.

A method for extracting an emitter with a known PRI is the Sequence Search Algorithm. This algorithm starts from a pulse and looks forward for another pulse at that PRI interval distance. This algorithm also allows missing pulses to provide continuation of the search.

The investigated algorithms have been implemented with MATLAB language and tests have been run on a Pentium 4 /. 2.80 GHz PC. For reliable performance analysis a receiver simulator has been developed. Using the input radar parameters, the simulator generates the output of a receiver which is the PDWs for the given scenario. Three example scenarios are generated to see the results of each technique. Example scenarios include radars difficult to identify, which are frequency hopping, frequency agile, PW agile or emitters with complex PRI types, such as staggered, dwell & switch and sinusoidal.

After the scenarios are created, the simulated PDWs are input to the deinterleaving algorithms. Example scenarios show that applying modifications improve the performance and ability of these algorithms. It is also observed that each algorithm gives better results for different scenarios.

The clustering algorithm deinterleaves emitters using frequency, AOA and PW parameters of pulses. Thus the clustering method is susceptible to variations in these parameters; however, this method can determine these parameters very accurately. The TOA based methods deinterleave emitters deriving the PRI of emitters from the TOA of pulses. Thus these methods are susceptible to variations in the PRI parameter, but they can estimate PRI accurately. It seems that a method utilizing both of these algorithms can give better results than each of these algorithms.

For full deinterleaving of all emitters, several schemas utilizing one or more of these algorithms are proposed together with some additional techniques to increase the reliability of these algorithms. The resultant schemas get a block of data received in a predefined time interval and utilize a clustering algorithm remembering previously determined clusters for pre-deinterleaving of emitters. Then known emitters are tracked in data collected in the corresponding clusters. The proposed method for tracking of known emitters is the Sequence Search Algorithm in both TOA and PA. Remaining data in each cluster is deinterleaved using one of the TOA based algorithms to detect a new emitter. A PA based method is combined with the deinterleaving stage to improve the reliability of detection.

These algorithms can be combined in many different ways. Therefore, nine different versions of deinterleaving schemas are developed. Three very difficult test scenarios are generated to experiment the resultant deinterleaving schemas. The test scenarios were really difficult, which include lots of frequency agile, frequency hopping, PW agile, jittered PRI, dwell& switch PRI, staggered PRI, sinusoidal PRI and sliding PRI emitters. In all scenarios radars are placed very close to each other in bearing.

Using the simulated PDWs for tests scenarios, proposed deinterleaving schemas are tested. The results of tests have shown that the schema with combination of wavelet based pulse detector and clustering technique gives the best results for deinterleaving task. This schema has detected 18 of 19 radars successfully in a very complicated scenario. However, this successful deinterleaving schema is four or five times slower than other algorithms.

The proposed schema has processed 200k PDWs in about 500 seconds for a simulation time of 20 seconds. By implementing this schema in C language it may be expected that the schema will work 25 times faster. With this assumption the simulated data can be processed in real time. Even with this performance assumption, the number of pulses processed per second is about 10k. Modern ESM systems process 500k pulses per second. However, by use of a specific hardware and software, the required performance may be achieved. Also the clustering and tracking stages can be parallelized with the deinterleaving stage of this schema to increase the performance.

In steady state all emitters in the environment will be known. Pulses coming from these emitters will be tracked, and no further deinterleaving will be needed until a new emitter enters the range of the EW system. This will cause an increase in the performance of the deinterleaving schema because the slowest part of this schema is the deinterleaving stage.

In real world missing pulses introduce a great problem for both deinterleaving and tracking algorithms so that the algorithms should be robust against them. Since the receiver simulator is designed to generate missing pulses, the proposed schemas are developed to be robust to them. Another problem related to the real world is the parameter tuning of algorithms. Since in the environment there are many different types of radars produced by different companies, tuning parameters of algorithms to successfully detect and track all emitters will be very difficult, even impossible. In the developed schema all of these problems are kept in mind and methods to provide reliable deinterleaving have been proposed. It is stated that the proposed schema is either suitable or can be configured to be used in real EW systems.

In the proposed schema, each hop of a frequency hopping radar and each PRI of a dwell & switch radar will be found as a single emitter. These emitters can be linked to be from the same radar by means of checking the other parameters of these emitters. Also for staggered PRI emitters a similar method is required, when frame PRI is detected.

In the proposed schema emitters may not directly correspond to a radar, but each emitter may be only corresponding to a mode of radar. All pulses of a radar, which may be tracked by one or more emitters, may be investigated by another program which only tries to find individual PRIs, assuming all pulses are received from a single radar.

Whenever all pulses coming from a single radar are extracted; beside calculated parameters mean frequency, AOA, mean PW, mean PRI and maximum PA, all other parameters such as individual PRIs, ASP, AST and BW can be determined via further processing of these pulses. Since all required parameters are determined the radar may easily be identified from the threat library.

A well known technique for tracking is Kalman Filter. As an alternative to the Sequence Search Algorithm, a Kalman Filter may be used in tracking stages of the proposed schemas. A problem with Kalman Filter may be missing pulses; then a method should be presented to take care of missing pulses.

The mother wavelet used in this study has a rectangular window. Thus the detector does not make any weighting on pulses. Effect of each pulse to the output of the detector is equal. Some other mother wavelets, which are weighting the pulses, can be studied and effect of mother wavelet may be examined.

The receivers in EW systems are not ideal; therefore, they produce some false PDWs. These extraneous pulses may affect the performance of the deinterleaving algorithm. In some cases the receiver's detection threshold is decreased to be able to detect emitters with low pulse amplitudes; but this causes an increase in false alarm rate. The receiver's false alarm rate can be simulated, and the performance of the deinterleaving algorithm for different false alarm rates can be studied.

For ELINT purposes the algorithm may also be run offline so that the processing speed will not be a critical problem. Generally ELINT systems try to calculate actual pulse parameters rather than only detection of a specific emitter.

In fact such a complete study may not be found in public literature because this task is closely related to military applications.

Although the proposed schema with wavelet detector has some disadvantages, like processing speed; it has lots of advantages over all other known methods and it is more successful. It is claimed that with a proper implementation this schema can be used in real EW systems.

REFERENCES

[1] Wiley, R. G., 1993. “ELINT: The Analysis of Radar Signals”, Artech House, Inc.

[2] Wiley, R. G., 1985. “ELINT: The Interception of Radar Signals”, Artech House, Inc.

[3] Avionics Department of the Naval Air Warfare Center Weapons Division, 1993, “Electronic Warfare and EW Systems”.

[4] Hertigian, J. A., 1975, “Clustering Algorithms”, John Wiley & Sons, Inc.

[5] Mardia H. K., 1988, “Adaptive Multi-dimensional Clustering for ESM”, IEE Colloquium on Signal Processing for ESM Systems.

[6] Mardia, H. K., 1989. “New Techniques for the Deinterleaving of Repetitive Sequences”, IEE Proceedings Radar and Signal Processing F 4, pp 149-154.

[7] Milojevic, D. J., and Popovic, B. M., 1992. “Improved Algorithm for the Deinterleaving of Radar Pulses”, IEE Proceedings Radar and Signal Processing F 1, pp 98-104.

[8] Orsi, R. C., Moore, J. B. and Mahony, R. E., 1996, “Interleaved Pulse Train Spectrum Estimation”, Proceedings of the Fourth International Symposium on Signal Processing and Its Applications, vol. 1, pp 125-128.

[9] Driscoll, D. E., Howard, S. D., 1999, “The Detection of Radar Pulse Sequences by Means of a Continuous Wavelet Transform”, Proceedings of IEEE International

Conference on Acoustics, Speech, and Signal Processing, vol. 3, pp 1389-1392.

[10] Rogers, J. A. V., 1985, “ESM Processor System for High Pulse Density Radar Environments” IEE Proceedings, vol. 132, pp 621-625.

[11] Logothetis, A., Krishnamurthy, V., 1998, “An Interval-Amplitude Algorithm for Deinterleaving Stochastic Pulse Train Sequences”, IEEE Transaction on Signal Processing, vol. 46, pp 1344-1350.

[12] Güven, E., 1994, “Emitter Identification by the Use of Clustering Techniques”, M. Sc. Thesis, METU.

[13] Güvenlik, A. R., 1999, “Clustering Techniques for Emitter Identification in Electronic Warfare”, M. Sc. Thesis, METU.

[14] Chandra, V., Bajpai, R.C., 1992, “ESM Data Processing Parametric Deinterleaving Approach”, Proceedings of IEEE International Conference on Computers, Communications and Automation, vol.1, pp 26-30.

[15] Kahrizi, M., Kabir, E., Bakhshi, H. R., 1997, “A New Algorithm for the Deinterleaving of Radar Pulses” International Journal of Engineering, vol. 10, pp 143-152.

[16] Frankpitt, B., Baras, J., and Tse, A., 2003 “A New Approach to Deinterleaving for Radar Intercept Receivers”, Proceedings of SPIE, vol. 5077, pp175-186.