NEW METHODS
FOR RADAR EMITTER IDENTIFICATION

Kenan GENÇOL

PhD Dissertation

Graduate School of Sciences

Electrical and Electronics Engineering Program

April, 2015

JÜRİ VE ENSTİTÜ ONAYI

Kenan Gençol' un "**New Methods for Radar Emitter Identification**" başlıklı **Elektrik-Elektronik Mühendisliği** Anabilim Dalındaki, Doktora tezi 03.04.2015 tarihinde aşağıdaki jüri tarafından Anadolu Üniversitesi Lisansüstü Eğitim - Ögretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

	Adı -Soyadı	**İmza**
Üye (Tez Danışmanı)	**: Yrd.Doç. Dr. Nuray AT**	**.....**
Üye	**: Prof. Dr. Ömer Nezih GEREK**	**.....**
Üye	**: Doç. Dr. Serkan GÜNAL**	**.....**
Üye	**: Doç. Dr. Rıfat EDİZKAN**	**.....**
Üye	**: Yrd. Doç. Dr. Mehmet KOÇ**	**.....**

Anadolu Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun

..... tarih ve ..... sayılı kararıyla onaylanmıştır.

Enstitü Müdürü

ABSTRACT

PhD Dissertation

NEW METHODS
FOR RADAR EMITTER IDENTIFICATION

Kenan GENÇOL

Anadolu University
Graduate School of Sciences
Electrical and Electronics Engineering Program

Advisor: Asst. Prof. Dr. Nuray AT
Co-advisor: Assoc. Prof. Dr. Ali KARA
2015, 90 pages

In this thesis new methods are introduced for radar emitter identification. Radar emitter identification is the process of identifying surrounding threat emitters in electronic warfare environments.

A method is developed for deinterleaving of radar pulse sequences. For this purpose, first clustering performances of two self-organizing neural networks, namely SOM and Fuzzy ART are evaluated. Then, a pulse amplitude tracking algorithm is proposed for dynamically varying signal environments wherein radar parameters can change abruptly. Simulation results show that the proposed algorithm can successfully deinterleave radar emitters that have agile pulse parameters.

Another method is developed for the recognition of pulse repetition interval modulation patterns. The method is based upon new features extracted from multiresolution wavelet decompositions of different types of pulse repetition interval modulation sequences. Simulation results show that recognition performance of the proposed features outperform conventional histogram based methods in both accuracy and computation time.

Keywords: Electronic Warfare, Electronic Support Measures, Pulse Deinterleaving, Pulse Repetition Interval Modulation, Multiresolution Wavelet Decomposition, Self-organizing Neural Networks.

ÖZET

Doktora Tezi

RADAR VERİCİLERİNİN KİMLİKLENDİRİLMESİ İÇİN YENİ METOTLAR

Kenan GENÇOL

Anadolu Üniversitesi
Fen Bilimleri Enstitüsü
Elektrik-Elektronik Mühendisliği Anabilim Dalı

Danışman: Yrd. Doç. Dr. Nuray AT
Eş Danışman: Doç. Dr. Ali KARA
2015, 90 pages

Bu tezde radar vericilerinin kimliklendirilmesine yönelik yeni metotlar geliştirilmiştir. Radar verici kimliklendirme elektronik harp ortamlarında etrafta tehdit oluşturan vericilerin kimliklendirilmesi sürecidir.

Geliştirilen metotlardan bir tanesi radar darbe dizilerinin ayrıştırılmasına yönelik geliştirilmiştir. Bu amaç için, öncelikle SOM ve Bulanık ART kendini örgütleyen sinir ağlarının kümeleme performansları değerlendirilmiştir. Daha sonra radar parametrelerinin ani olarak değişebildikleri dinamik değişken sinyal ortamları için bir darbe genliği izleme algoritması geliştirilmiştir. Benzetim sonuçları önerilen algoritmanın çevik darbe parametrelerine sahip radar vericilerini başarılı bir şekilde ayrıştırabildiğini göstermektedir.

Bir diğer metot darbe tekrarlama aralığı modülasyon örüntülerinin tanınması üzerinedir. Metot farklı tip darbe tekrarlama aralığı modülasyon dizilerinin çoklu çözünürlüklü dalgacık ayrıştırmasıyla çıkarılan yeni özniteliklere dayanmaktadır. Metot çok geniş bir aralıktaki darbe tekrarlama aralığı modülasyon parametreleriyle test edilmiştir. Benzetim sonuçları önerilen özniteliklerin oldukça gürbüz olduklarını ve geleneksel histogram tabanlı metotları hem doğruluk hem de hesaplama süreleri yönünden geçtiklerini göstermektedir.

Anahtar Kelimeler: Elektronik Harp, Elektronik Destek Sistemleri, Darbe Ayrıştırma, Darbe Tekrarlama Aralığı Modülasyonu, Çoklu çözünürlüklü Dalgacık Analizi, Kendini Örgütleyen Sinir Ağları.

ACKNOWLEDGEMENTS

I would like to express my special thanks to my advisor Dr. Nuray At and co-advisor Dr. Ali Kara for their guidance and help throughout the thesis study. I also would like to thank members of the thesis committee Dr. Ömer N. Gerek and Dr. Serkan Günal for their valuable comments in this study.

I would like to express my deepest gratitude to my family for their love, support, and patience.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AOA	: Angle of Arrival
CF	: Carrier Frequency
DOA	: Direction of Arrival
DTOA	: Difference of Time of Arrival
ECM	: Electronic Countermeasures
ECCM	: Electronic Counter-Countermeasures
EDV	: Emitter Description Word
ELINT	: Electronic Intelligence
ESM	: Electronic Support Measures
EW	: Electronic Warfare
PA	: Pulse Amplitude
PDV	: Pulse Description Word
PF	: Pulse Frequency
PRI	: Pulse Repetition Interval
PW	: Pulse Width
RF	: Radio Frequency
RWR	: Radar Warning Receiver
TOA	: Time of Arrival

To my parents

"*Education is what remains after one has forgotten everything he learned in school.*"

Albert Einstein (*may be misattributed*)

1. INTRODUCTION

In dense electronic warfare environments, a large number of radar emitters can be active simultaneously. Radar emitter identification is called the process of identifying surrounding threat emitters in electronic warfare environments. This process is generally carried out as follows:

A radar intercept receiver which is a passive receiver picks up an interleaved stream of pulses in natural time of arrival order and extracts their pulse parameters. These parameters are sent to a deinterleaving subsystem which sorts them and forms pulse cells that each are assumed to belong to a specific emitter. Then these grouped pulse parameters are sent to a pulse repetition interval (PRI) modulation subsystem which recognizes the pulse repetition interval modulation type of potential emitters. This recognition task is crucial for identification of the emission source and its mission for possible counter measures. Finally, potential emitters are identified according to emitters stored in the threat library. A block diagram of the process is shown in Figure 1.1.

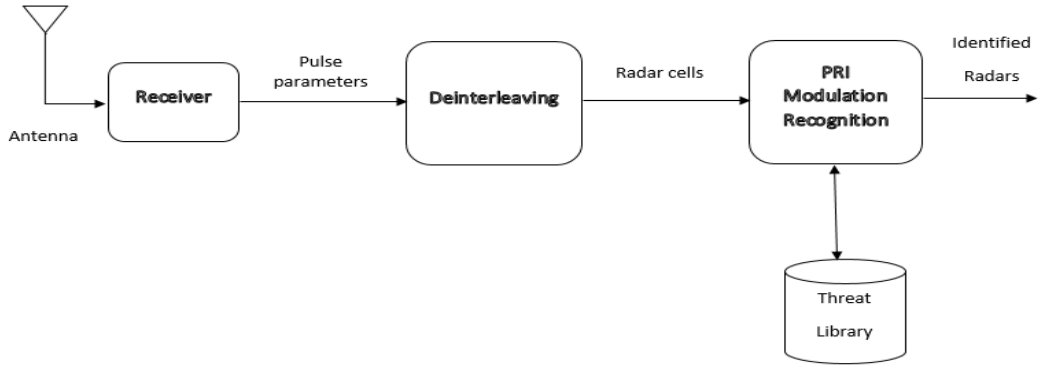

Figure 1.1. Radar emitter identification

This thesis is devoted to two main phases of radar emitter identification: deinterleaving and PRI modulation recognition. The rest of the thesis is organized as follows:

In Chapter 2, some preliminary concepts about radar emitter identification are discussed. The chapter begins with the definition of electronic warfare and then describes the function of an electronic support measures system and general characteristics of emitters accordingly. Some of the emitter characteristics such as pulse frequency, pulse width, pulse amplitude and angle of arrival are discussed in detail. Definitions of PRI modulation types which is basis for Chapter 4 are also given here. The chapter finalizes with the problem of emitter identification and some earlier methods in literature that have been proposed to solve this problem.

In Chapter 3, the problem of deinterleaving is considered. Beyond the conventional histogram and clustering techniques, clustering task is achieved by employing self-organizing neural networks for this purpose. Clustering performances of two self-organizing neural networks, namely Self Organizing Maps (SOM) and Fuzzy-ART are evaluated in terms of clustering quality, computation time, convergence time and algorithmic complexity in detail and the results are presented. Then, a pulse amplitude tracking algorithm is developed to improve deinterleaving capability in dynamically varying multi-function radar environments.

In Chapter 4, the problem of recognizing different pulse repetition interval modulation patterns is considered. For this purpose, features based on the multi-resolution wavelet analysis of PRI modulation patterns are proposed. The performance of the proposed feature set are evaluated in terms of accuracy and computation time and the results are presented. The current version of this chapter has been accepted for publication [44].

Finally, some concluding remarks about the thesis are presented in Chapter 5.

2. BACKGROUND INFORMATION

2.1. Electronic Warfare

The concept of Electronic Warfare (EW) is to use electromagnetic spectrum to determine enemy's order of battle, intensions and capabilities or to prevent hostile use of electromagnetic spectrum [1].

Basically EW systems include Electronic Intelligence (ELINT), Electronic Support Measures (ESM), Radar Warning Receiver (RWR), and Electronic Counter Measures (ECM) systems. The major difference between ELINT and ESM systems is their response time. ELINT systems provide a form of detailed measurement for signal analysis, whereas ESM systems provide a real time projection of radar activity. RWRs indicate whether the platform is under attack. RWR may be a part of an active system which applies ECM to a hostile system [1].

2.2. Electronic Support Measures

Electronic support measures (ESM) system performs the functions of threat detection and area surveillance to determine the identity of surrounding emitters.

Broadly speaking, an automated ESM system consists of three main subsystems as shown in Figure 2.1. The receiver subsystem, which is a passive radar receiver, picks up the pulses transmitted by various radars in the environment and measures their individual parameters. Basically, these parameters include angle of arrival (AOA), pulse frequency (PF), pulse width (PW), pulse amplitude (PA), and time of arrival (TOA). The receiver is designed to cover wide parameter ranges to ensure detection of all radars of interest. The measured parameters of every successfully intercepted pulse are encoded in digital format called pulse description word (PDV) [2].

The deinterleaver sorts the PDVs and forms pulse cells, each comprises a set of PDVs assumed to belong to the same emitter. Then each pulse cell is encoded as emitter description word (EDV) [2].

The main processor compares the estimated EDVs with stored data in threat library of EW system in order to identify the intercepted radar type or to update the emitters stored in the threat library. In some advanced EW systems, the main processor can further determine the location of the detected radar emitter from identified radar type, instantaneous position data supplied from airborne navigator, angle of arrival information, Electronic Order of Battle (EOB) stored in the threat library of the EW system. The output of the ESM system is fed to an ECM resource manager system, which starts an appropriate ECM action against the intercepted radar. Thus, if some of the radar cells generated at the output of the deinterleaver are not related to actual radars in the environment (false radars), this will lead to wasteful use of military resources against these false radars [2].

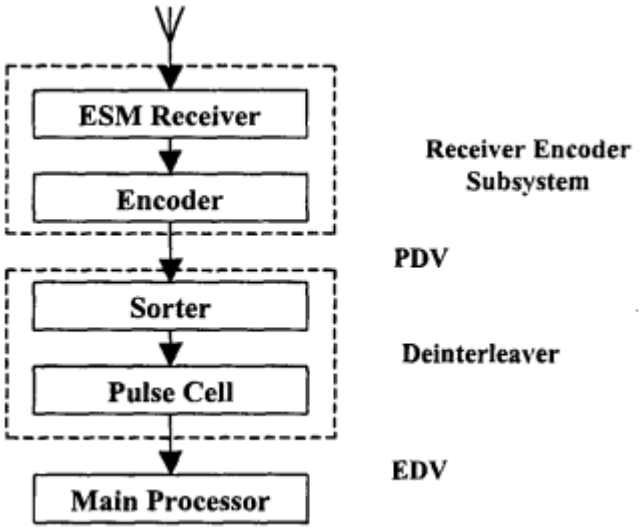

Figure 2.1. Basic layout of an ESM system [2]

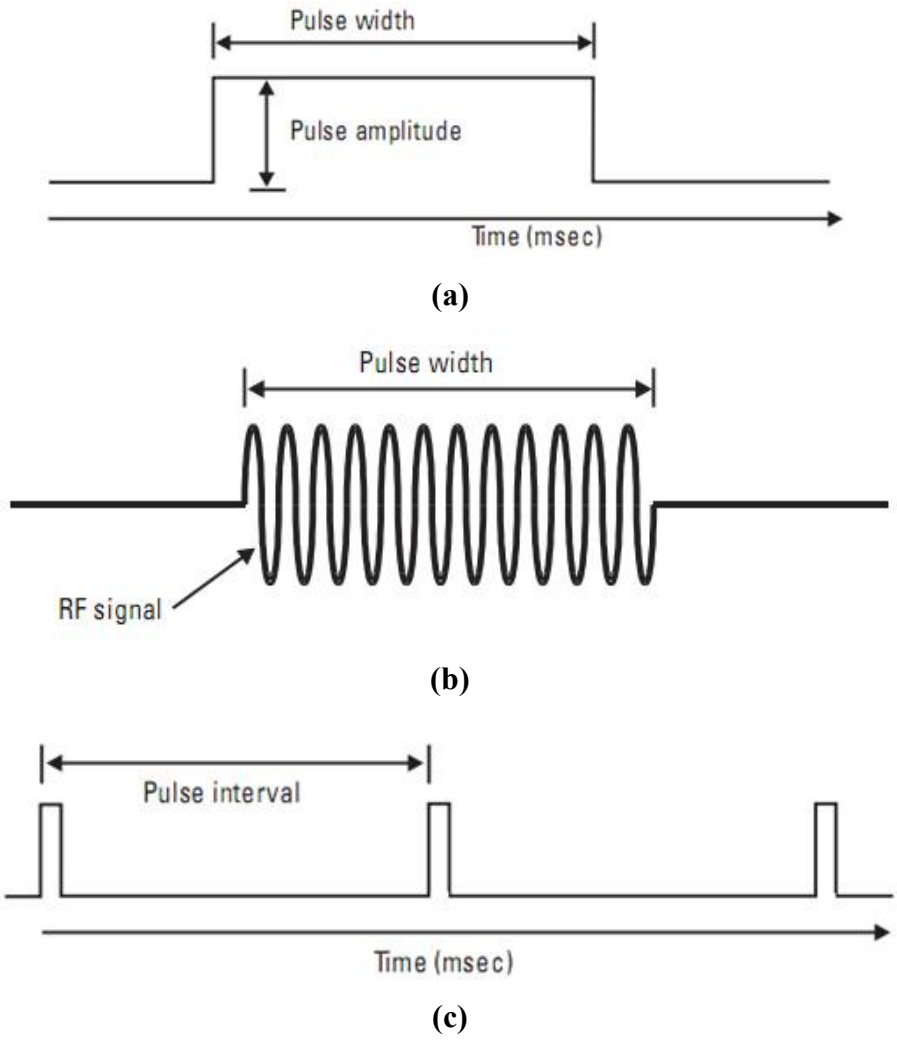

Figure 2.2. (a)-(c) Pulse modulation parameters (b) Transmitted pulse signal [3]

2.3. Emitter Characteristics

In order to determine the identity of surrounding emitters, some parameters of the received radar pulses must be measured that is characterized by emitters (Figure 2.2). These parameters include the following:

- Pulse Frequency (PF)
- Pulse Width (PW)
- Pulse Amplitude (PA)
- Angle (direction) of Arrival (AOA) (DOA)

- Time of Arrival (TOA)
- Pulse Repetition Interval (PRI) (derived)
- PRI Type
- Antenna Scan Type and Rate (derived)
- Lobe Duration (Beam Width) (derived)

2.3.1. Pulse frequency (PF)

In many of pulse parameter clustering applications, it is emphasized that the carrier frequency is the next most important parameter after AOA. The major advantage of frequency parameter is highlighted if we notice the fact that radars physically near to each other cannot operate on the same frequency [4]. One of the major problems in using carrier frequency data for deinterleaving is the increasing use of *frequency agility* or *frequency hopping* by the modern emitters [5]. Frequency agility usually refers to the radar's ability to change the transmitter frequency on a pulse-to-pulse basis randomly. The frequency of the next pulse cannot generally be predicted from the frequency of the current pulse. Frequency hopping is the capability to operate at a number of frequencies on a pulse group-to-pulse group basis and to change frequencies in minimal time periods (longer than a few PRIs) [6]. Modern emitters can randomly hop from one frequency to another frequency within its frequency hop set, or randomly change its frequency within a frequency bandwidth which is called agile frequency. By doing this, emitters try to apply ECCM against jamming threats directed to them [5].

2.3.2. Pulse width (PW)

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 [1].

Pulse width is accepted as a less effective clustering parameter due to the fact that many RADARs are similar in this respect and also multipath situations may cause variations in measured PW values [4]. In addition to this, 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].

2.3.3. Pulse amplitude (PA)

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 [1].

At the first glance, PA can be seen an unreliable parameter for clustering because of its variability within a pulse train due to antenna scanning. However, the amplitude change from one pulse to the next in a pulse train is not so great. It is almost certain that the adjacent pulses with a large amplitude difference do not come from the same emitter [6]. In addition to this, its usage with Time of Arrival (TOA) information can reveal the scan pattern of the RADAR which can identify the threat type of the emitter.

Since PA is a strong function of distance to the emitter, it can easily deinterleave signals coming from the emitters located far away from each other [4].

2.3.4. Angle of arrival (AOA)

The azimuth angle of arrival is the direction in which the pulse is received. Although it is possible for an emitter to vary its other parameters from pulse to pulse, it would however have to move at enormous speeds to change its bearing during that interval [5]. Even if RADAR or a receiving system is moving, this

movement will be very slow as compared to the pulse rate [1]. It is a fact that even an airborne RADAR cannot change its location in a few milliseconds of the PRI time, so the AOA measurement by an intercept receiver is considered as relatively stable one [4].

2.3.5. Time of arrival (TOA)

A measured time of arrival parameter is related to pulse repetition interval (PRI) parameter of radars. PRI is defined as the difference of time of arrival of consecutive pulses. PRI determines radar's maximum unambiguous range and maximum unambiguous velocity [1, 6].

Radars can use more than one PRI and can change their PRI values in very different types called PRI modulation. Radars may change their PRI modulation to resolve ambiguities or to improve their ECCM capabilities [1, 6].

2.3.6. Common PRI modulation types

Common PRI modulation types are given below [7] and illustrated in Figure 2.3.

Constant (Stable) PRI

The radar has a nearly constant PRI if the peak variations are less than about 1% of the mean PRI. Such variations are considered incidental in that they generally serve no useful purpose.

Jittered PRI

Intentional PRI variations are used for a variety of radar purposes. The radar has large variations – for example, up to about 30% of the average PRI.

Dwell and Switch PRI

The radar has bursts of pulses with several stable PRIs switched from one burst to the next.

Stagger PRI

Stagger PRI is the use of two or more PRIs selected in a fixed sequence. The sequence may contain more than one of the several intervals before it repeats. The sequence is described by the number of ‘‘positions’’ used to make up one period of the sequence.

Sliding PRI

A sliding PRI is characterized by monotonic increases or decreases in the PRI followed by a rapid switch upon reaching one extreme limit to return to the other extreme limit.

Periodic PRI

Periodic PRI modulation is a nearly sinusoidal variation over a more limited range than sliding PRI.

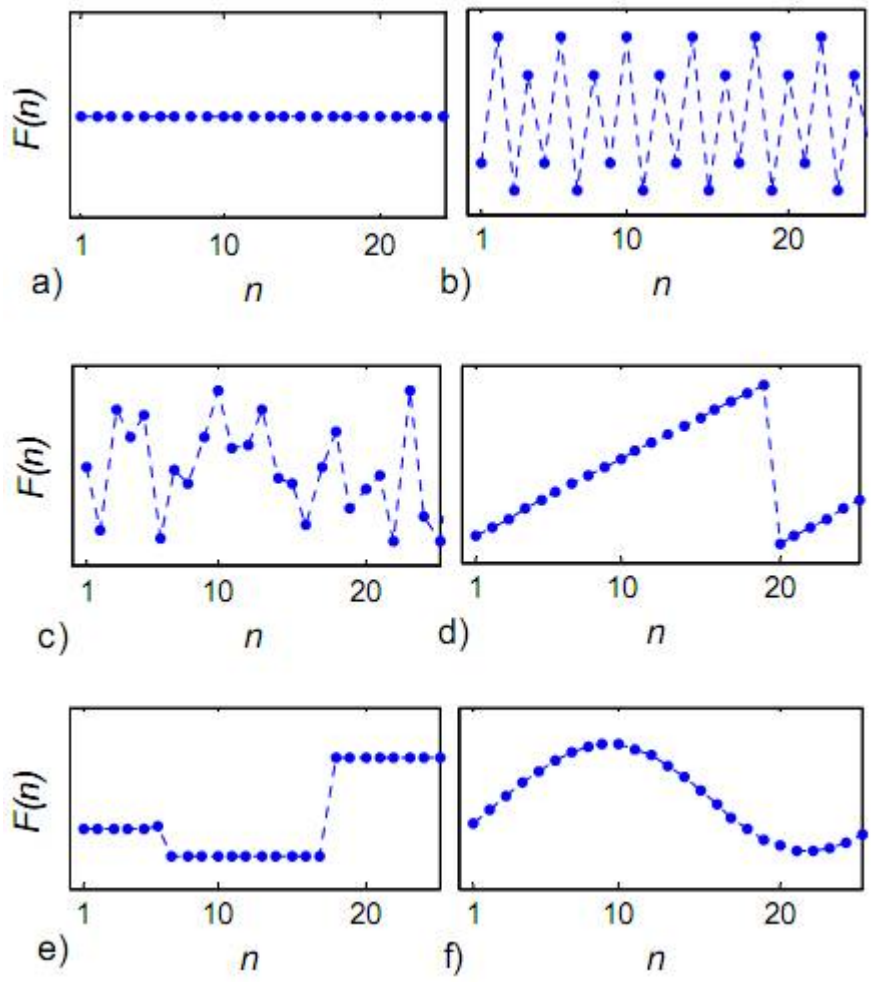

Figure 2.3. Some examples of common PRI modulation types: (a) Constant (b) Stagger (c) Jittered (d) Sliding (e) Dwell &Switch (f) Periodic (*n* and *F(n)* denote time index and PRI, respectively) [8]

2.4. ESM Receiver

ESM systems are much more sophisticated than simple RWRs and are equipped with many auxiliary circuits. A possible ESM configuration with blocks characterizing various equipment modules is shown in Figure 2.4.

Since frequency measurement is a costly process, it is performed only in an independent channel whose antenna is able to cover the whole area and all frequency bands to be explored. The antenna is omnidirectional with a horizontal 360-degree beam; the beam is either a dedicated beam or is synthesized from the direction-finding (DF) channels.

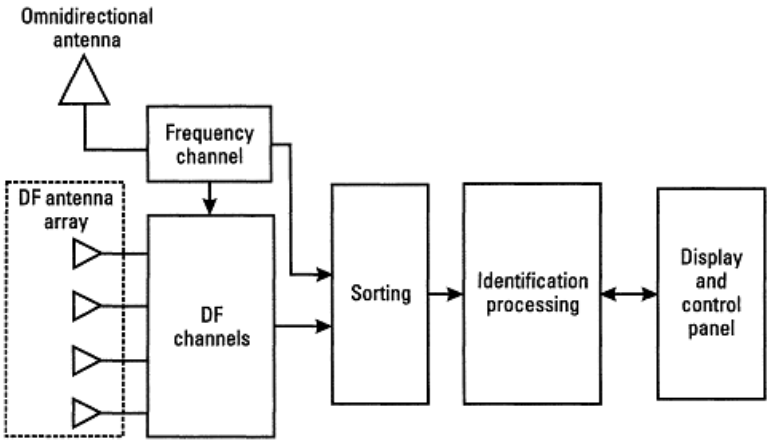

Figure 2.4. Block diagram of a typical ESM system [9]

Detection of signal activity, frequency measurements, and sometimes PWs are generally derived from this antenna channel [9].

The DF receiver generally consists of *n* simple channels, each consists of a directional antenna. Sometimes the gain of these antennas is exploited to avoid costly amplification before the main receiver. Comparison of the amplitudes received from different channels determines the direction of origin of a signal. Pulse characteristics, translated into digital messages specifying TOA, PW, frequency, DOA, and amplitude, are sent to the sorting system to be segregated into trains [9].

A high-speed computer, or fast dedicated hardware, subdivides the pulses into groups whose similar characteristics indicate that they probably come from the same emitter (sorting or preprocessing). This computer performs a preliminary analysis of the data. A second computer more slowly correlates the different emitters detected by the sorter, determines the operational mode of each emitter (scanning or tracking), computes the antenna scan periods (ASP), and possibly identifies the emitter by comparing its parameters with those memorized in the library. The data is then shown on a display [9].

2.4.1. The radar scan

The radar scan (to an EW receiver) is the time history of the received signal strength. This is formed by the shape of the radar antenna beam and its angular movement relative to the location of the EW receiver. Figure 2.5 shows the antenna gain pattern of a radar antenna in polar coordinates (in one dimension). The antenna beam is shown as rotating (in that dimension) relative to an EW receiver location. Note that the main beam and side lobes all rotate past to the EW receiver. Figure 2.6 shows the relative amplitude of the signal received by the EW receiver as a function of time. The shape of this curve can be analyzed to determine the beamwidth and scanning pattern of the radar [10].

The ground-based search radar antenna will typically rotate 360° in azimuth (a circular scan). This causes an EW receiver to see evenly spaced main beams as shown in Figure 2.7.

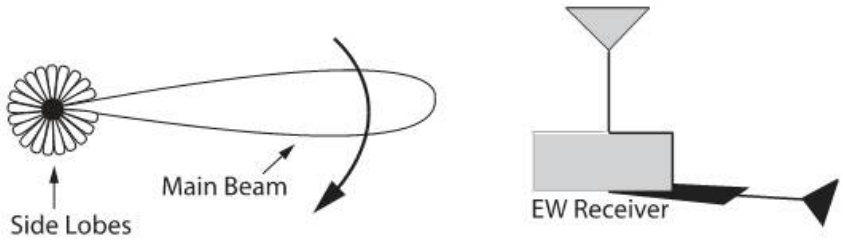

Figure 2.5. The narrow radar beam moves past to the location of an EW receiver, illuminating it with its main beam and its side lobes [10]

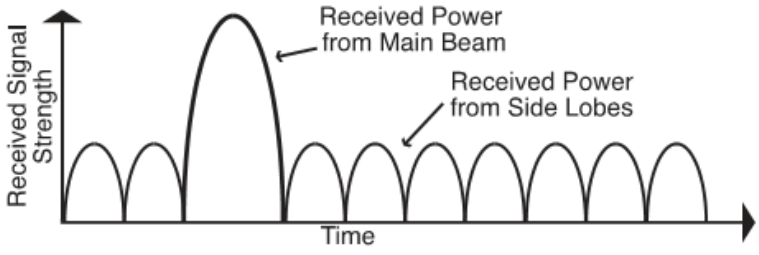

Figure 2.6. The EW receiver observes the rotating-antenna beam as a time history of the received signal strength of the threat radar [10]

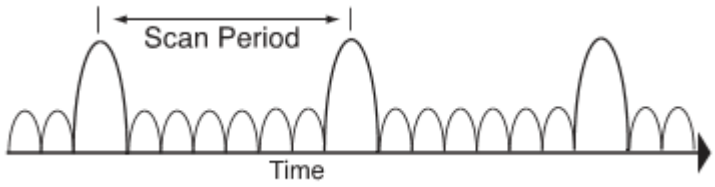

Figure 2.7. In a circular scan, the time interval between two receptions of the main beam is equal to the antenna-scan period [10]

The time between main beams is equal to the period of rotation [10]. The radar typically has a narrow antenna beam which allows it to determine the azimuth and elevation of a target. The more accurately the radar must know the location of the target, the narrower the beam. The cross-range dimension of a radar's resolution cell is generally determined by the 3-dB antenna beamwidth as shown in Figure 2.8 [10].

2.4.2. Radar antenna

The parabola, shown in Figure 2.9, works well as a reflector of electromagnetic energy and has been the basis for many radar antennas. The parabolic surface is illuminated by a source of radiated energy called the feed, which is placed at the

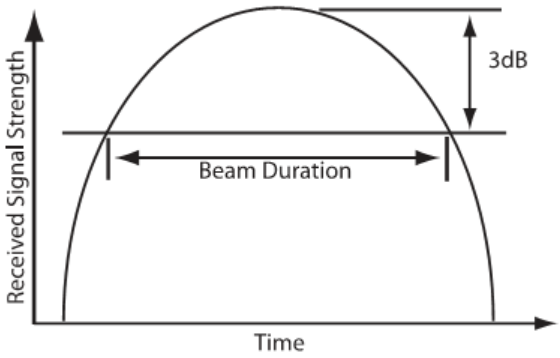

Figure 2.8. If the rotation rate of the antenna can be determined, the beamwidth can be derived [10]

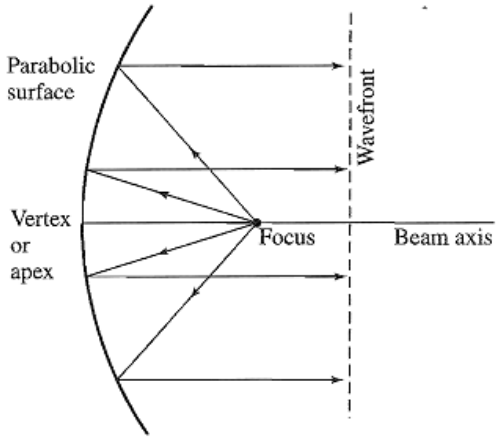

Figure 2.9. Contour of a parabolic reflector antenna [11]

focus of the parabola. The parabola converts the spherical wave radiated from the feed to a plane wave [11].

2.4.3. Antenna radiation pattern

At microwave frequencies, the most common feeds are rectangular, circular, or corrugated horns. For a rectangular aperture, the radiation pattern simplifies into two principal planes (azimuth and elevation in radar case) [12]:

$$f(\theta,\varphi) = \frac{\sin(\theta)}{\theta}.\frac{\sin(\varphi)}{\varphi} \tag{2.1}$$

where θ and φ denote angle in azimuth and elevation planes, respectively. Figure 2.10 shows the radiation pattern in one plane:

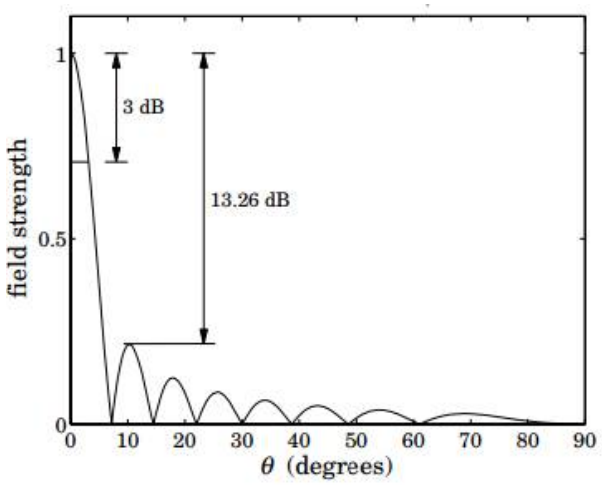

Figure 2.10. Typical radiation pattern [12]

2.5. Emitter Identification

The problem of emitter identification is a rather longtime problem in EW systems. Earlier methods use two main approaches to solve this problem: Clustering, and TOA deinterleaving techniques. Also, there are some other methods that solve emitter identification problem in general and deinterleaving problem in particular. All these methods are discussed in the following subsections.

2.5.1. Clustering techniques

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. Clustering techniques [13] are based on similarities and differences of stationary emitter parameters like PF, PW, and AOA. In some limited circumstances, PA can also be used. This gives us a chance to reveal other parameters and thus identify the emitter [1].

2.5.2. TOA deinterleaving techniques

TOA deinterleaving techniques are based on difference of time of arrival (DTOA) histograms. All TOA deinterleaving algorithms make a coarse search through histograms to find out PRIs.

All differences histogram (A-DIF)

This deinterleaving algorithm [7] is based on the DTOA histogram of all consecutive pulses. 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. Then the histogram of this all differences data is generated. However, in practice, only the differences up to a certain level can be used [1].

Cumulative differences histogram (C-DIF)

This algorithm [14] is an improvement to the A-DIF algorithm. The distinction is that the calculation of all differences at the beginning is not necessary. 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 A-DIF 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 in this fashion [1].

Sequential differences histogram (S-DIF)

An improvement to the C-DIF algorithm is the sequential differences histogram algorithm [15]. 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 C-DIF algorithm threshold used was proportional to 1/PRI, whereas it is proved that the optimum threshold for the sequential differences histogram is proportional to e^{-PRI}. The second peak required in the C-DIF algorithm is not used in this method, because only one level difference is available [1].

2.5.3. Sequence search method

A "sequence search" is another method for deinterleaving [14]. 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 [1].

2.5.4. PRI transform

This method [16] utilizes an algorithm that is auto correlation-like integral which leads to a kind of PRI spectrum, wherein the locations of the spectral peaks indicate the PRI values. This is called PRI transform.

2.5.5. Other methods

One of the methods [17] is based on Kalman filtering theory. This method formulates the pulse train deinterleaving problem as a stochastic discrete-time dynamic linear model. The deinterleaving detection/estimation task is done by using Kalman filter.

Another method [18] is based on neural networks where a vector neural network (VNN), a supervised learning algorithm, is applied to emitter identification problem.

Also, there are some methods based on the spectrum estimation/detection of pulse trains. One approach [19] is to search periodogram for pulse repetition frequencies. Another approach [20] utilizes a continuous wavelet transform for detecting characteristic period or scale, T of radar pulse sequences.

3. DEINTERLEAVING

3.1. Introduction

The main purpose of Electronic Support (ES) systems is to intercept as many electromagnetic emissions as it can, deinterleave them and thus identify the surrounding threatening emitters in the environment. ES systems primarily rely on algorithms based on PRI analysis to deinterleave received pulse sequences [14-16, 21, 22]. But, due to incarnation of a broad range of complex PRI patterns in evolving modern electronic warfare environments that spurred an increasing complexity in these algorithms, it has become an indispensable task to assist this deinterleaving process [24]. Thus, clustering algorithms that depend on individual parameters of radar pulses such as pulse frequency, pulse width and angle of arrival are needed. In literature, studies in this field are quite limited and in one of them [23], a clustering algorithm is developed based only on intrinsic pulse shape without making use of signal parameters mentioned above. In [24], authors consider self-organizing neural networks for fast clustering of radar pulses. But in this study, the employed radar data set is quite limited and gives a general idea about performances of neural networks by utilizing some other well-known data sets together.

ES systems generally encounter with a continuous stream of pulses accompanied by many imperfections such as noisy signals, time of arrival uncertainty, missing and spurious pulses. Furthermore, to handle such circumstances they should work in a real-time basis. Thus, clustering algorithms that are suitable for ES systems should have the following requirements. First and most important of all, they should not have prior knowledge about the number and features of categories formed. Second, due to the high density of pulses received, they should process them sequentially without storing in long terms. Lastly, they should be implemented in hardware in a fast and parallelized manner. For such requirements, self-organizing neural networks which lend themselves to highly parallelized autonomous clustering seem very promising [24].

This study improves the work presented in [24]. The clustering performances of two self-organizing neural networks, namely Self Organizing Maps (SOM) and Fuzzy-ART are evaluated in terms of clustering quality, computation time, convergence time and algorithmic complexity in detail and the results are presented. Then, a pulse amplitude (PA) tracking algorithm is proposed to assist deinterleaving of radar pulses in dynamically varying multi-function signal environments where emitters have agile pulse parameters.

3.2. Performance Evaluation of Self-Organizing Neural Networks

3.2.1. Self-organizing maps (SOM) (Kohonen network)

The material presented here is based on [25, 26]:

The SOM consists of a regular, usually two-dimensional (2-D), grid of map units (Figure 3.1). Each unit i is represented by a prototype vector, $m_i = [m_{i1} \ldots m_{id}]$ where d is input vector dimension. The units are connected to adjacent ones by a

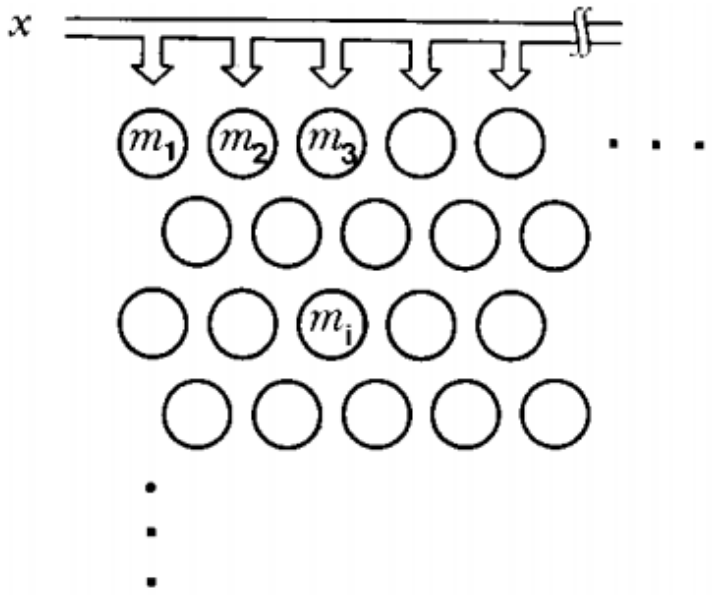

Figure 3.1. SOM structure [26]

neighborhood relation. The number of map units, which typically varies from a few dozen up to several thousand, determines the accuracy and generalization capability of the SOM. During training, the SOM forms an elastic net that folds onto the "cloud" formed by the input data. Data points lying near each other in the input space are mapped onto nearby map units. Thus, the SOM can be interpreted as a topology preserving mapping from input space onto the 2-D grid of map units.

The SOM is trained iteratively. At each training step, a sample vector $\boldsymbol{x}$ is randomly chosen from the input data set. Distances between $\boldsymbol{x}$ and all the prototype vectors are computed. The best-matching unit (BMU), which is denoted here by b, is the map unit with prototype closest to $\boldsymbol{x}$.

$$\|\boldsymbol{x} - \boldsymbol{m}_b\| = \min_i\{\|\boldsymbol{x} - \boldsymbol{m}_i\|\} \tag{3.1}$$

Next, the prototype vectors are updated. The BMU and its topological neighbors are moved closer to the input vector in the input space. The update rule for the prototype vector of unit i is

$$\boldsymbol{m}_i(t+1) = \boldsymbol{m}_i(t) + \alpha(t)h_{bi}(t)[\boldsymbol{x} - \boldsymbol{m}_i(t)] \tag{3.2}$$

where t denotes time, $\alpha(t)$ represents adaptation coefficient, and $h_{bi}(t)$ denotes neighborhood kernel centered on the winner unit given by

$$h_{bi}(t) = exp\left(-\frac{\|\boldsymbol{r}_b - \boldsymbol{r}_i\|^2}{2\sigma^2(t)}\right) \tag{3.3}$$

where $\boldsymbol{r}_b$ and $\boldsymbol{r}_i$ are positions of neurons b and i on the SOM grid, respectively. Both $\alpha(t)$ and $\sigma(t)$ decrease monotonically with time. There is also a batch version of the algorithm where the adaptation coefficient is not used. In the case of a discrete data set and fixed neighborhood kernel, the error function of SOM is given by

$$E = \sum_{i=1}^{N} \sum_{j=1}^{M} h_{bj} \left\| \boldsymbol{x}_i - \boldsymbol{m}_j \right\|^2 \tag{3.4}$$

where N is the number of training samples, and M is the number of map units. Neighborhood kernel h_{bj} is centered at unit b, which is the BMU of vector $\boldsymbol{x}_i$, and evaluated for unit j.

3.2.2. Fuzzy ART

The material presented here is based on [27]:

ART Field Activity Vectors: Fuzzy ART system includes a field F_0 of nodes that represent a current input vector; a field F_1 that receives both bottom-up input

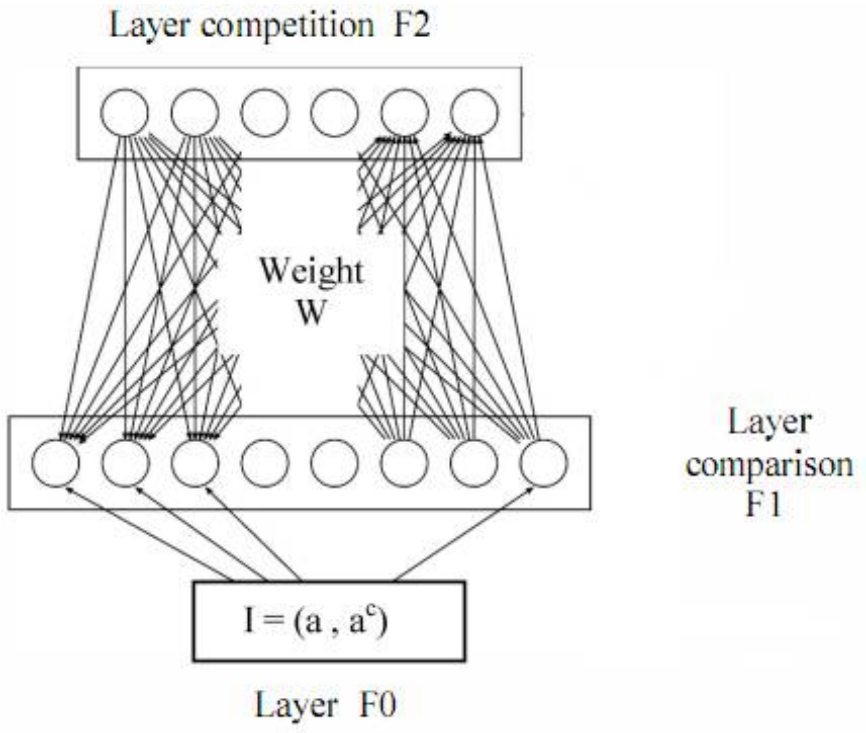

Figure 3.2. Fuzzy ART Network [27]

from F_0 and top-down input from a field F_2 that represents the active code, or category (Figure 3.2). The F_0 activity vector is denoted by $\boldsymbol{I} = (I_1 \ldots I_M)$ with each component I_i in the interval [0, 1], $i = 1 \ldots M$. The F_1 activity vector is denoted by $\boldsymbol{x} = (x_1 \ldots x_M)$ and the F_2 vector is denoted by $\boldsymbol{y} = (y_1 \ldots y_N)$. The number of nodes in each field is arbitrary.

Weight vector: Associated with each F_2 category node j ($j = 1 \ldots N$), there is a vector $\boldsymbol{w}_j = (w_{j1}, \ldots, w_{jM})$ of adaptive weights. Initially,

$$w_{j1}(0) = \cdots = w_{jM}(0) = 1; \tag{3.5}$$

then each category is said to be *uncommitted*. After a category is selected for coding it becomes *committed*. Each weight w_{ji} is monotonically non-increasing and hence converges to a limit. The Fuzzy ART weight vector $\boldsymbol{w}_j$ subsumes both the bottom-up and top-down weight vectors of ART.

Parameters: Fuzzy ART dynamics are determined by a choice parameter $\alpha > 0$; a learning rate parameter $\beta \in [0, 1]$; and a vigilance parameter $\rho \in [0, 1]$.

Category choice: For each input $\boldsymbol{I}$ and category j, the choice function, T_j, is defined by

$$T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{\alpha + |\boldsymbol{w}_j|} \tag{3.6}$$

where the fuzzy AND [28] operator $\wedge$ is defined by

$$(\boldsymbol{x} \wedge \boldsymbol{y})_i \equiv \min(x_i, y_i) \tag{3.7}$$

and the l_1 norm $|.|$ is used:

$$|\boldsymbol{x}| \equiv \sum_{i=1}^{M} |x_i| \tag{3.8}$$

For notational simplicity, $T_j(\boldsymbol{I})$ in (3.6) is often written as T_j when the input $\boldsymbol{I}$ is fixed. The category choice is indexed by J, where

$$T_J = \max\{T_j : j = 1 \ldots N\} \tag{3.9}$$

If more than one T_j is maximal, the category j with the smallest index is chosen. In particular, nodes become committed in order j *(1, 2...N)*

Resonance or reset: *Resonance* occurs if the match function of the chosen category meets the vigilance criterion; that is, if

$$\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_J|}{|\boldsymbol{I}|} \geq \rho \tag{3.10}$$

Learning then ensues. On the other hand, *Mismatch reset* occurs if

$$\frac{|\boldsymbol{I} \wedge \boldsymbol{w}_J|}{|\boldsymbol{I}|} < \rho \tag{3.11}$$

The value of the choice function T_J is reset to -1 for the duration of the input presentation to prevent its persistent selection during search. A new index J is chosen according to (3.9). The search process continues until the chosen J satisfies (3.10).

Learning: The weight vector $\boldsymbol{w}_J$ is updated as follows

$$\boldsymbol{w}_J^{(new)} = \beta\left(\boldsymbol{I} \wedge \boldsymbol{w}_J^{(old)}\right) + (1 - \beta)\boldsymbol{w}_J^{(old)} \tag{3.12}$$

Fast learning corresponds to setting $\beta = 1$.

Fast-commit-slow-recode option: For efficient coding of noisy input sets, it is useful to set $\beta = 1$ when J is an uncommitted node, and then take $\beta < 1$ after the category is committed. Then $\boldsymbol{w}_J^{(new)} = \boldsymbol{I}$ is the first time category J becomes active.

Input normalization option: Proliferation of categories is avoided in Fuzzy ART if inputs are normalized; that is, for some $\gamma > 0$:

$$|\boldsymbol{I}| = \gamma \tag{3.13}$$

for all inputs $\boldsymbol{I}$. Normalization can be achieved by preprocessing each incoming vector $\boldsymbol{a}$, i.e. setting

$$\boldsymbol{I} = \frac{\boldsymbol{a}}{|\boldsymbol{a}|} \tag{3.14}$$

An alternative normalization rule, called *complement coding*, achieves normalization while preserving amplitude information. Complement coding represents both the on-response and the off-response to $\boldsymbol{a}$. To define this operation in its simplest form, let $\boldsymbol{a}$ represent the on-response. The complement of $\boldsymbol{a}$, denoted by $\boldsymbol{a}^c$, represents the off-response, where

$$a_i^c = 1 - a_i \quad i = 1, \dots, M \tag{3.15}$$

The complement coded input $\boldsymbol{I}$ to the recognition system is the $2M$-dimensional vector

$$\boldsymbol{I} = (\boldsymbol{a}, \boldsymbol{a}^c) = (a_1, \dots, a_M, a_1^c, \dots, a_M^c) \tag{3.16}$$

Note that

$$|\boldsymbol{I}| = |(\boldsymbol{a}, \boldsymbol{a}^c)| = \sum_{i=1}^{M} a_i + \left(M - \sum_{i=1}^{M} a_i\right) = M \tag{3.17}$$

so inputs preprocessed into complement coding form are automatically normalized. In complement coding, the initial condition (3.5) is replaced by

$$w_{j1}(0) = \cdots = w_{j,2M}(0) = 1 \tag{3.18}$$

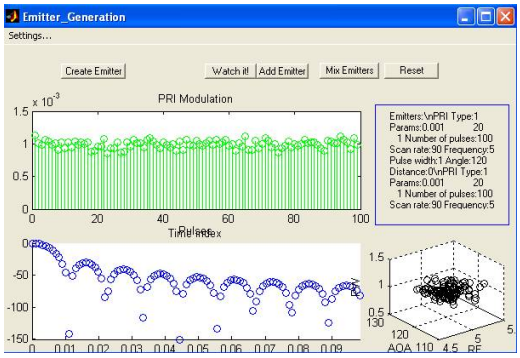

Figure 3.3. Synthetic mixed pulse generator

3.2.3. Simulations

Modeling environments consisting of multiple emitters is important for realistic implementation of an ESM receiver that integrates radar pulses and processes them. For this reason, a so-called synthetic mixed pulse generator (simulator) is designed. A screen output of the designed simulator is given in Figure 3.3. Detailed description of the simulator is presented in Appendix A.1.

Test Scenarios

Test scenarios presented here is based on [1, 5, and 6]. Three different scenarios are tested to see the clustering quality, computation time, and convergence time of two self-organizing neural networks. Scenarios are given in Table 3.1. There are 10, 11 and 19 radars in each scenario, respectively. In each scenario, a Gaussian noise of zero mean and 5% standard deviation, which is an extreme case, is added to AOA, PF, and PW values to simulate the measurement errors. Each scenario is simulated for one second, and a total number of 5260, 22690, and 75050 pulses which is ordered in natural time of arrival are obtained, respectively.

Table 3.1. Radar Parameters of Test Scenarios

	Radar Parameters											
	Scenario 1				**Scenario 2**				**Scenario 3**			
Radar No	**AOA (Degree)**	**PF (MHz)**	**PW (µs)**	**PRF (Hz)**	**AOA (Degree)**	**PF (MHz)**	**PW (µs)**	**PRF (Hz)**	**AOA (Degree)**	**PF (MHz)**	**PW (µs)**	**PRF (Hz)**
1	80	7500	20	500	50	9375	3	1800	30	4500	7	215.0
2	75	4300	30	350	60	9000	2	5000	50	4500	3	300 ± 6 (jittered)
3	70	2700	10	400	60	9350	2	2700	50	5500	7	475.0
4	85	1800	25	550	65	3100	2	800	40	4000	3	170.0
5	65	1300	12	750	70	3000	3	750	35	3000	15	800.0
6	78	9000	40	1000	70	5000	2	2200	20	8400	11	220.0 ± 30 (sinusoidal)
7	73	2200	35	650	75	9500	1	1500	45	3000	7	200.0 / 450.0 / 750.0 / 980.0 (dwell & switch)
8	60	3800	22	250	80	3250	1	3940	65	3000	7	1100.0 / 1115.0 / 1150.0 / 1180.0 (staggered)
9	67	5500	15	300	90	3200	10	1000	5	3000	7	175.0
10	60	9300	50	350	100	9000	2	3000	45	3000	15	800.0
11	80	2700	5	160					5	4000	3	145.0 ± 4.8 (jittered)
12									30	4000	3	105 ± 4.2 (jittered)
13									55	5100	11	100.0 / 250.0 / 550.0 / 550.0 (staggered)
14									85	5100	11	200.0 / 450.0 / 750.0 / 980.0 (dwell & switch)
15									70	6800	80	440 to 580

												(sliding)
16									70	8400	11	220.0 ± 30.0 (sinusoidal)
17									70	6600	20	335.0
18									85	8100	15	1050.0
19									30	9800	1	80.0

3.2.4. Results and Discussion

For each scenario, SOM and Fuzzy ART clustering results are given in Table 3.2 and Table 3.3, respectively. It is be observed that SOM algorithm can reach to best and stable clustering accuracies, with 92.1%, 96.0% and 95.0% in each scenario, respectively. The sacrifice for such good clustering accuracies is the relatively more computation and longer convergence times (Table 3.2). SOM networks, where map sizes obtained from Principal Component Analysis (PCA) analysis are also given in the last rows of each scenario tab. The clustering accuracies for this case are 83.3%, 94.7%, and 93.5% respectively, which are quite close to best clustering quality scores. But, if the computation and convergence times are compared, these PCA based SOM networks have very less computation and fast convergence times compared to “best” SOM networks counterparts as we call (SOM networks that have best accuracies - emphasized by bold letters in each scenario tab). This is due to the fact that instead of selecting SOM inputs in random manner, inputs are first selected among the samples that lie on the principal axis obtained from PCA analysis. This selection criterion has a significant impact on the computation and convergence times and gives the optimum performance bounds of SOM networks. But, it also should not have been forgotten that overall data is needed for PCA analysis, so it is rather uncommon situation for online clustering of data which is the case for ESM systems.

Table 3.2. SOM Clustering Results

SOM Network For Scenarios	Map Size (MxN)	Clustering Quality (%)	Computation Time (sec)	Convergence Time (epochs)	Topographic Error [0,1)
Scenario 1 ($2 \geq r(0) \geq 1$, $0.05 \leq \alpha(t) \leq 0.5$)	5x5	39.3	0.27	$I \leq 15$	0.002
	6x6	56.6	0.61	$I \leq 20$	0.000
	8x8	79.0	1.20	$I \leq 36$	0.004
	10x10	80.1	1.89	$I \leq 50$	0.000
	15x15	**92.1**	**6.16**	**$I \leq 120$**	**0.021**
	12x9 (PCA)	83.3	2.67	$I \leq 62$	0.002
Scenario 2 ($3 \geq r(0) \geq 1$, $0.05 \leq \alpha(t) \leq 0.5$)	4x4	38.9	1.14	$I \leq 44$	0.000
	5x5	59.1	1.23	$I \leq 44$	0.000
	6x6	78.4	1.24	$I \leq 44$	0.020
	10x10	89.6	2.47	$I \leq 67$	0.010
	20x20	**96.0**	**14.06**	**$I \leq 226$**	**0.053**
	18x13 (PCA)	94.7	7.29	$I \leq 124$	0.037
Scenario 3 ($4 \geq r(0) \geq 1$, $0.05 \leq \alpha(t) \leq 0.5$)	5x5	55.4	4.48	$I \leq 160$	0.015
	10x10	82.3	6.50	$I \leq 178$	0.072
	15x15	91.2	11.80	$I \leq 242$	0.070
	20x20	94.0	20.48	$I \leq 325$	0.035
	25x25	**95.0**	**36.23**	**$I \leq 490$**	**0.052**
	25x18 (PCA)	93.5	22.23	$I \leq 298$	0.066

Table 3.3. Fuzzy ART Clustering Results

Fuzzy ART Network For Scenarios	Vigilance Number (ρ)	Clustering Quality (%)	Computation Time (sec)	Weight (MxN)	Convergence Time (epochs)
Scenario 1 (β=1, α=0.001)	0.75	70.08	0.62	6x6	$I \leq 2$
	0.80	69.91	0.64	6x9	$I \leq 2$
	0.85	89.63	0.72	6x12	$I \leq 2$
	0.90	90.49	0.80	6x11	$I \leq 2$
	0.92	90.596	0.78	6x15	$I \leq 2$
	0.88	**90.52**	**0.72**	**6x10**	**$I \leq 2$**

Scenario 2 (β=1, α=0.001)	0.75	79.34	1.86	6x5	I ≤ 2
	0.80	78.85	2.15	6x8	I ≤ 2
	0.85	82.99	2.16	6x8	I ≤ 2
	0.90	84.41	2.53	6x12	I ≤ 2
	0.95	90.12	3.17	6x20	I ≤ 2
	0.92	**90.19**	**2.30**	**6x10**	**I ≤ 2**
Scenario 3 (β=1, α=0.001)	0.75	76.19	10.23	6x16	I ≤ 2
	0.80	76.57	9.97	6x16	I ≤ 2
	0.85	82.38	14.73	6x31	I ≤ 2
	0.90	**89.52**	**15.66**	**6x31**	**I ≤ 2**
	0.82	80.65	13.03	6x24	I ≤ 2
	0.92	87.58	19.25	6x44	I ≤ 2

On the other hand, Fuzzy ART networks have relatively low best clustering accuracies which are about 90%, they have the best computation time and convergence time scores (Table 3.3). For example, PCA based SOM networks have computation time scores of 2.67, 7.29, and 22.23 seconds (Table 3.3) whereas Fuzzy ART networks that have best clustering accuracies in each scenario have corresponding computation scores of 0.72, 2.30, and 15.66 seconds (Table 3.3), respectively. In addition to this, all Fuzzy ART networks work in fast learning mode and they have convergence time of just 2 epochs.

Although SOM networks have relatively high and stable clustering accuracies, predefined map size, more computation and longer convergence times, and difficulty in determining the category bounds can be considered the major drawbacks of these networks. On the other hand, while Fuzzy ART networks have less computation times and they converge fast, relatively low clustering quality scores and proliferation of categories in high vigilance numbers become their bottlenecks. In addition, both networks have the algorithmic complexity of $O(MN)$ where M is the number of samples and N is the number of neurons. However, this N number which determines the number of clusters is large in SOM networks compared to their Fuzzy ART counterparts.

If clustering accuracy, computation time and convergence rate criteria are considered altogether, Fuzzy ART networks seem more promising algorithm for online clustering of interleaved radar data.

3.3. Methodology

3.3.1. Problem statement

Multi-function radars (MFRs) which can perform several tasks simultaneously by changing the pulse frequency and pulse width are very common in today's modern electronic warfare environments. Radar's capability to change the frequency on a pulse-to-pulse basis is called *frequency agility*, and on a pulse group-to-pulse group basis is called *frequency hopping*. Modern emitters can randomly hop from one frequency to another frequency within its frequency hop set, or randomly change its frequency within a frequency bandwidth which is called agile frequency.

This multi-function operation leads to some heterogeneous clusters in conventional pulse deinterleaving process where pulses belonging to a specific emitter class are grouped within other emitter clusters. This misleading causes incorrect clustering of radar data and directly affects the subsequent processing performed on data.

For a specific emitter, the amplitude change from one pulse to the next in a pulse train is not so great. It is almost certain that the adjacent pulses with a large amplitude difference do not come from the same emitter. In addition to this, its usage with Time of Arrival (TOA) information can reveal the scan pattern of the RADAR which can identify the threat type of the emitter.

3.3.2. Modeling of pulse amplitude values at the receiver side

Let $e_1, e_2, \ldots e_N$ be the N emitters surrounding our ESM system. Let w_i be the circular scan rate of ith emitter. Then the principal angle (direction) of its main beam (beam axis) at any time t is given as:

$$\theta_i(t) = \omega_i t \ (\text{mod } 360 \text{ deg}) \tag{3.19}$$

If emitter antennas are assumed to have uniform rectangular apertures, the strength of radiation in decibels (dB) emitted by an emitter is given in the canonical form of:

$$P(\theta) = 20log\left|\frac{sin\theta}{\theta}\right| \tag{3.20}$$

Then the magnitude of the pulse emitted from ith emitter at time t and angle θ is:

$$P_i(t,\theta) = P\big(\theta_i(t)\big) = 20log\left|\frac{\sin(\theta_i(t))}{\theta_i(t)}\right| \tag{3.21}$$

If the emitter is positioned at distance R_i from ESM receiver and emitting at frequency f_i, the path loss associated by the emitter in free space is given as:

$$L_{p,i} = 20\log\left(\frac{4\pi R_i}{\lambda_i}\right) \tag{3.22}$$

where $\lambda_i = c / f_i$, and $c = 3\text{x}10^8$ is the speed of light in free space in meters.

Then the magnitude of the pulse received from ith emitter at time t is derived as:

$$A_i(t) = P_i(t,\theta) - L_{p,i} = 20\log\left\{\left|\frac{\sin(\theta_i(t))}{\theta_i(t)}\right| / \left(\frac{4\pi R_i}{\lambda_i}\right)\right\} \tag{3.23}$$

In this formula, time t is a function of pulse repetition interval of the emitter. If the emitter is assumed to have a simple PRI value, then time of arrival of nth pulse received is

$$t = n.PRI_i \tag{3.24}$$

3.3.3. PA tracking algorithm

1. Sort the TOA's of the corresponding pulses in the cluster in ascending order.

2. Let

   A_i: $A_1, A_2 \ldots A_N$ denote the actual (received) PA values, and

   $\hat{A}_i$:$\hat{A}_1, \hat{A}_2, \ldots, \hat{A}_N$ denote the estimated PA values for each

   θ_{est}: initial estimate of scan angle $\theta = wt \bmod 2\pi$.

3. Recursively:

   3.1. Calculate the estimated $\hat{A}_i$'s by using Eq. (3.23),

   3.2. Find θ_{est} such that $\sum_{i=1}^{N} \left\| A_i - \hat{A}_i \right\|^2$ is minimized, i.e,

$$\theta = \min_{\arg \theta_{est}} \sum_{i=1}^{N} \left\| A_i - \hat{A}_i \right\|^2 \tag{3.25}$$

This is a nonlinear least squares estimation problem which can be solved by using Gauss-Newton method iteratively. Gauss-Newton method is a naïve method that best fits to model-based parameter estimation problems. By using Gauss-Newton method, an estimation problem is reduced to calculation of Jacobians of the residual function under consideration.

Here, the residual function is

$$r_i = A_i - \hat{A}_i \triangleq A_i - \frac{sin\theta_{est}}{\theta_{est}} \quad i = 1,2, \ldots, N \tag{3.26}$$

and, the Jacobian matrix of r_i ,

$$\boldsymbol{J} = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{\theta}_{est}} = \frac{cos\boldsymbol{\theta}_{est}}{\boldsymbol{\theta}_{est}} - \frac{sin\boldsymbol{\theta}_{est}}{\boldsymbol{\theta}_{est}^{\,2}} \tag{3.27}$$

Solve

$$\boldsymbol{J}^T\boldsymbol{r}\boldsymbol{J} = \left|\frac{\partial \boldsymbol{J}}{\partial \boldsymbol{\theta}_{est}}\right|\boldsymbol{r} \tag{3.28}$$

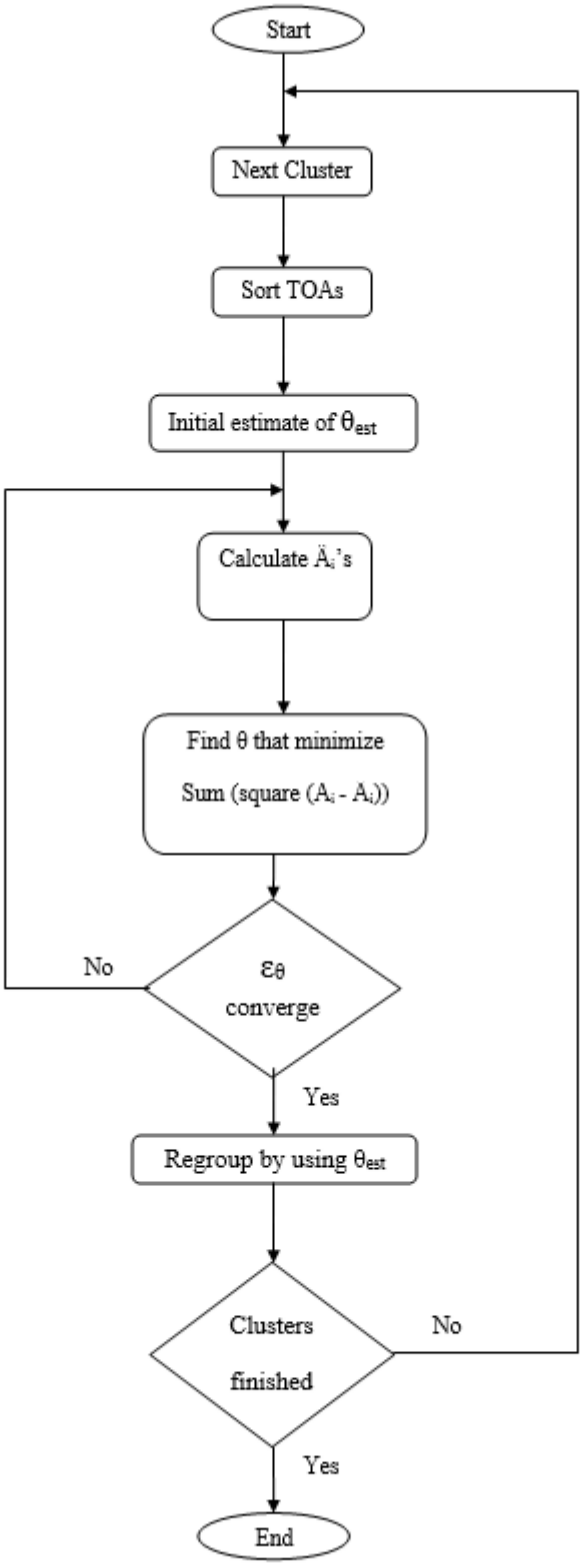

Figure 3.4. Flowchart of the proposed algorithm

until the difference between two subsequent θ_{est}, ε_{θ} converges to a choice value.

5. Regroup pulse cells by using present pulse parameters and new estimated emitter parameter, the scan rate of the emitter.

The flowchart of the proposed algorithm is given in Figure 3.4.

3.3.4. Illustrative examples

Two illustrative examples are presented to describe in which situations the proposed method works. In each example, a TOA noise of 1% is added to each TOA values to simulate timing jitter.

The first example illustrates a situation where emitters have the same PRF value but different scan rates. In this example, for simplicity, there are two emitters with the same PRF value 2 KHz and scan rates 30 and 45 deg/s, respectively. The normalized scan pattern of mixed emitters after simulation of 200 ms is illustrated in Figure 3.5.

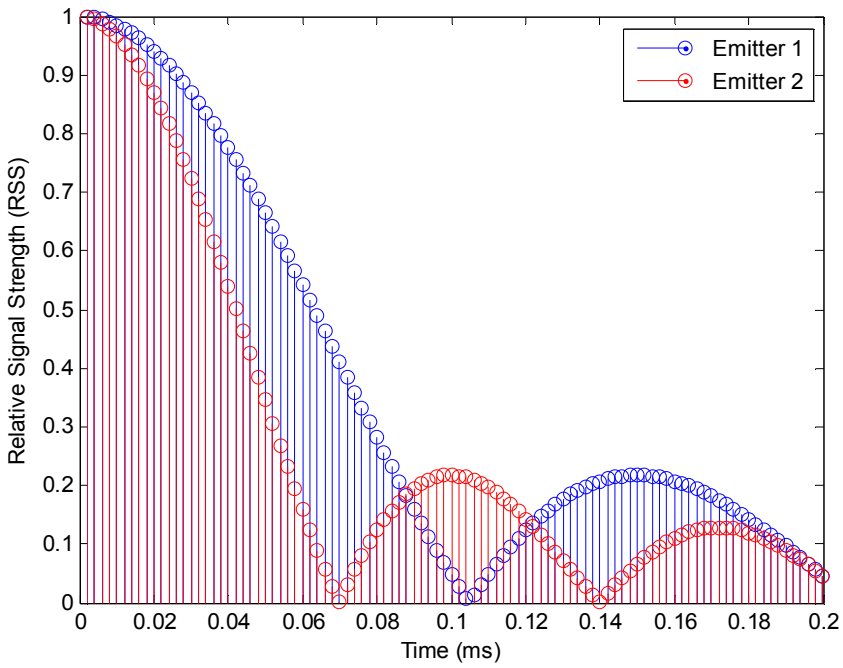

Figure 3.5. Two emitters with the same PRF but different scan rates

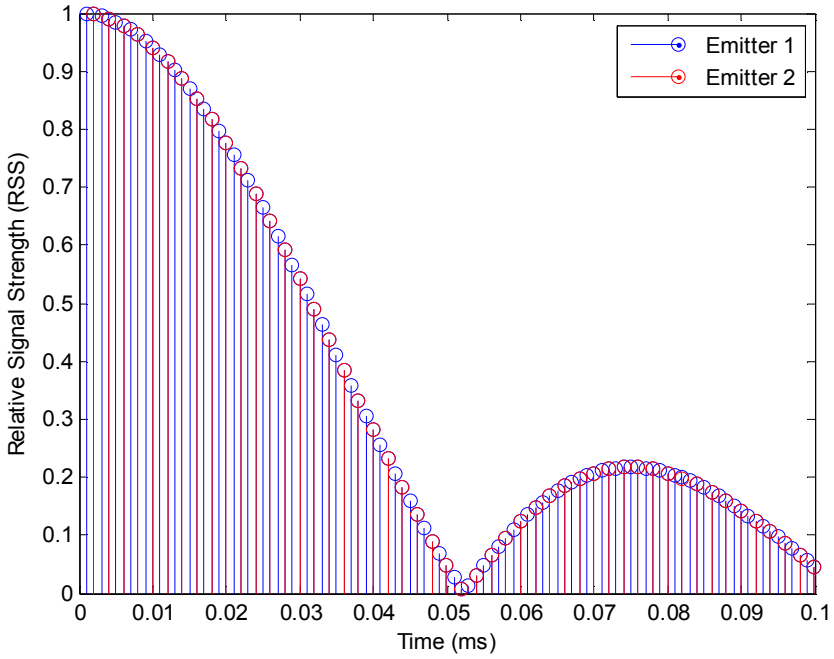

Figure 3.6. Two emitters with the same scan rate but different PRF values

The second example illustrates a situation where emitters have different PRF values but the same scan rate. In this example, for simplicity, there are two emitters with the same scan rate, 60 deg/s and PRF values 2 KHz and 3 KHz, respectively. The normalized scan pattern of mixed emitters after simulation of 100 ms is illustrated in Figure 3.6.

3.3.5. Convergence of Gauss-Newton method to the proposed model and some practical considerations

The Jacobian matrix of the model function has a pitfall. It is undefined at time instants where the derivative of the function changes sign. If these points are included in the calculation of the Jacobian matrix, the method converges very slowly, or does not converge at all. Thus, only the local regions between two

Table 3.4. Gauss-Newton estimation of scan angles of some emitters ranging from 45 deg/s to 180 deg/s

Actual angle	Initial estimate	Number of iterations	Estimated angle
45	120	5	93.09, 63.21, 47.20, 45.23, 45.01
90	120	3	94.98,90.66,90.07
135	120	2	134.71,134.99
180	120	2	181.18, 179.99

consecutive extremes (regions between a maximum – a minimum and a minimum – a maximum point) of the model function are considered.

Radar emitters in dynamically varying signal environments such as air-to-air electronic warfare environments have typically scan rates of 8 to 30 rpm. This corresponds approximately between 45 deg/s and 180 deg/s scan angles. Gauss-Newton estimation of scan angles of some emitters ranging from 45 deg/s to 180 deg/s after each iteration is given in Table 3.4. It estimates the scan rate of the emitters with a resolution of 0.1 deg/s by at most 5 iterations.

If we would apply bisection method to estimate scan rate of emitters between 45 and180 deg/s, it would take 2 iterations in the best case and 14 iterations in the worst case to have a resolution of 1 deg/s.

3.3.6. Simulations

Monte Carlo simulations are performed with aircraft radars which have multi-function operations. The radar data set is presented in Table 3.5. The radars operate between 8-12 GHz bands and have scan rates ranging from 8 rpm to 30 rpm. These radars also have agile PRF and agile pulse width values. 22 emitters are obtained by using the lower and upper bounds of PRF and PW values of each radar mode.

Table 3.5. The radar data set [11]

Radar Mode	PRF (KHz)	Pulse width (μs)
Range-Gated High PRF	100-300	1-3
Medium PRF	6-20	1-20
Burst Ranging	3-20	2-60
Active Track	8-300	0.1-20
Raid Assessment	2-16	2-60
Non Coop. Target Rec.	2-20	1-200
HiPwr Jam	50-300	3-10
Cal/AGC	2-300	0.1-60
Air Data Link	10-300	1-20
Gun Ranging	10-20	0.1-0.5
Weather Avoidance	0.5-5	1-50

3.3.7. Results and Discussion

Number of emitter clusters formed and the clustering quality of each cluster after 100 runs for Fuzzy ART (FA) and the proposed Fuzzy ART plus PA tracking (FA + PA) algorithms are given in Figure 3.7 and Figure 3.8, respectively. It is shown in Figure 3.7 that in each run, the number of clusters formed by FA + PA algorithm is closer to actual number of clusters, which is 22 than FA. Also, in each run the clustering quality of FA + PA is higher than FA. The average number of clusters formed and the average clustering quality of FA and FA + PA algorithms are presented in Table 3.6. The average number of clusters formed by FA is 15 with an average quality of 86.37%. On the other hand, FA + PA gives 18 clusters in average with an average clustering quality of 92.84%.

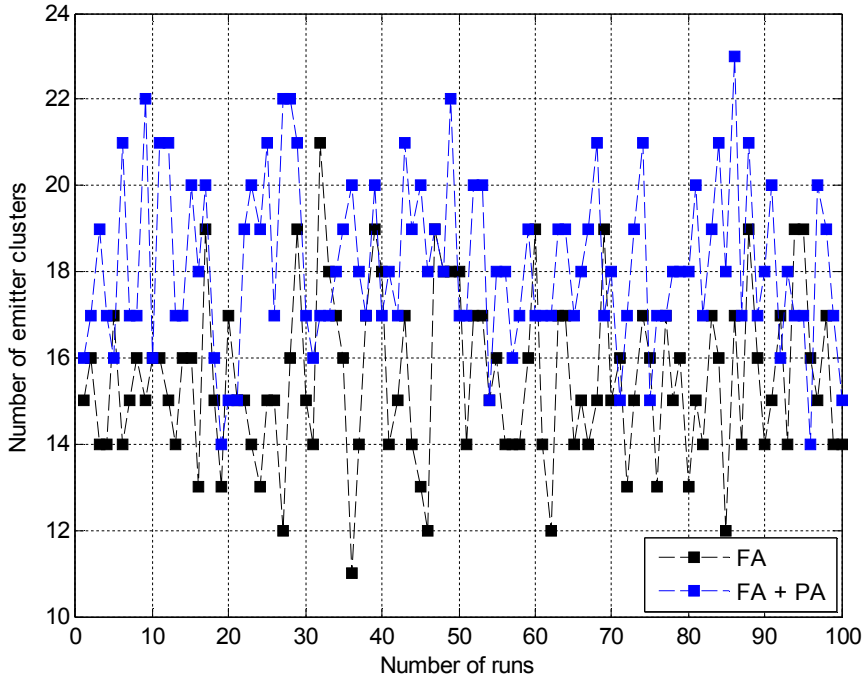

Figure 3.7. Number of emitter clusters formed in simulations

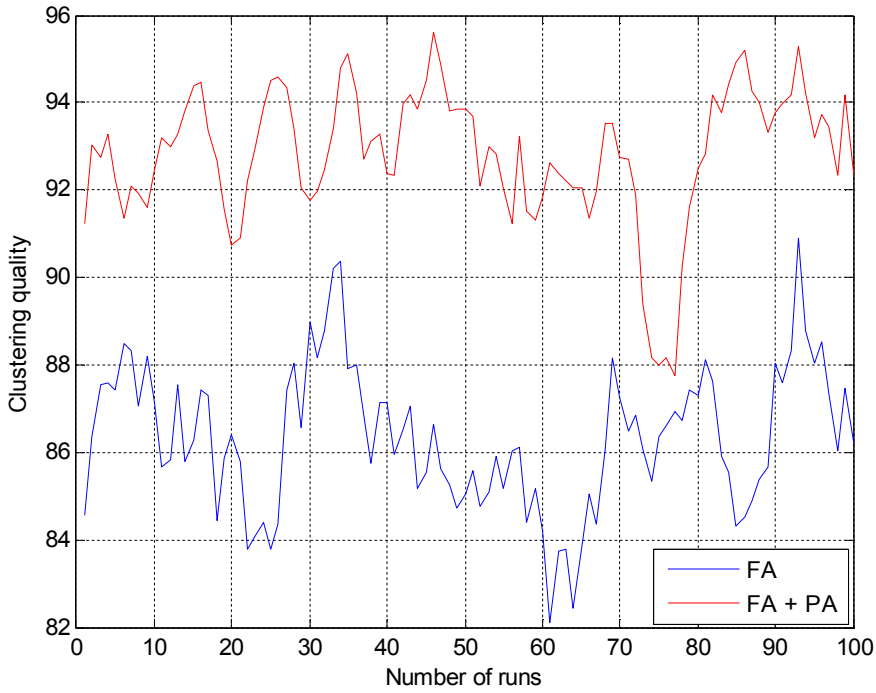

Figure 3.8. Clustering quality in simulations (smoothed data)

Table 3.6. Average clustering results

Clustering Algorithm	Average number of clusters formed	Average clustering quality (%)
FA	15	86.37
FA + PA	18	92.84

Simulation results for the case where emitters are positioned at the same direction is presented in Table 3.7. In this case, FA has an average clustering quality of 75.90%, whereas FA + PA has an average clustering quality of 85.42%.

It can be inferred from simulation results that clustering quality is improved when PA Tracking algorithm is employed.

Table 3.7. Average clustering results (emitters are positioned at the same direction)

Clustering Algorithm	Average clustering quality (%)
FA	75.90
FA + PA	85.42

4. PRI MODULATION RECOGNITION

4.1. Introduction

In dense signal environments where large number of (many) emitters can be active simultaneously, a radar intercept system receives an interleaved stream of pulses in natural time of arrival order. It is then the task of the intercept system to deinterleave this mixed pulse sequence and thus to identify the source emissions. For this identification task, various parameters such as pulse frequency (PF), pulse width (PW), pulse amplitude (PA), angle of arrival (AOA), and time of arrival (TOA) are measured and emission sources are classified accordingly. Among these pulse parameters, TOA is of considerable interest since it leads to a key derived parameter called Pulse Repetition Interval (PRI), which represents the difference of sequential TOAs of received pulses. Any emission source either intentionally or unintentionally varies (or modulates) this parameter for a specific mission requirement. Thus, it is important to recognize PRI modulations for identification of the emission source and its mission for possible counter measures. Additionally, some emitters may vary even PRI modulation type according to its mission. This makes estimation of PRI modulations more critical from operational point of view. As the subject is sensitive, and may require mostly classified data, it is not possible to see so many works toward resolving this problem.

4.2. Literature Survey

In the literature, deinterleaving pulse trains has been in the focus of many researches in the past years [14-22]. Several studies were performed on estimation of PRI to construct better deinterleaving algorithms. Cumulative Differences (C-DIF) and Sequential Differences (S-DIF) histogram techniques, techniques based on TOA matrix characteristics, and a transformation called PRI Transform leading to a kind of PRI spectrum were some amongst these

algorithms that enabled new methodologies to estimate pulse repetition intervals of pulse sequences and thus to deinterleave them [14-16,21]. But, as the signal environments become more complex in electronic warfare due to evolving of technology, the need for not only estimating pulse repetition interval of radar pulses, but also recognizing the modulation patterns hidden inside them has become an inevitable task. So, in recent years, studies have been focused on this area and several methods have been proposed to recognize PRI modulation types [29-36]. In Noone [29], an *N* dimensional feature vector is deduced by using the second differences of the TOAs of a pulse train and PRI modulation types are classified via a neural network. In Rong et al. [30], a two dimensional feature vector is formed by extracting frequency and shape features from this *N* dimensional vector which reduced the computation time greatly. In Ryoo et al. [31] PRI modulation types are recognized based on the features extracted from the autocorrelation of the PRI sequences for each PRI modulation type. In this method, due to the sensitivity of the features against signal imperfections, compensation of missing pulses and the removal of spurious pulses must be performed as a preprocessing step. In Kauppi et al. [32] PRI modulation patterns are classified hierarchically. First, six modulation patterns are first grouped into three sub-patterns by using a neural network classifier and then they are binary classified by using one-dimensional classifiers. Some proposed features in this method are based on sequential difference (SDIF) histograms [15] and they need to be calculated for several orders due to unknown signal parameters.

4.3. PRI Modulation

4.3.1. Definition

Let *F* be a function describing the PRI modulation type:

$$x_n = y_{n+1} - y_n = F(n) \quad n = 1,2,\dots,N-1 \tag{4.1}$$

where y_n is the time of arrival (TOA) of the nth pulse received in a pulse sequence of length N and x_n is the difference of TOAs of two consecutive pulses.

4.3.2. PRI modulation types

In general, there are six common PRI modulation types: stable, jittered, stagger, dwell & switch, sliding and periodic. Each modulation type serves for a specific purpose, thus they represent some characteristics of emitters. Common PRI modulation types are described below [7]:

Constant (Stable) PRI

The radar has a nearly constant PRI if the peak variations in PRI values are less than about 1% of the mean PRI. Such variations are considered incidental in that they generally serve no useful purpose. Constant PRI radar is defined by

$$F(n) = c, \;\; n = 2, \dots, N-1 \tag{4.2}$$

where c denotes a constant real number.

Stagger PRI

Stagger PRI is the use of two or more PRIs selected in a fixed sequence. The sequence may contain more than one of the several intervals before it repeats. The sequence is described by the number of ‘‘positions’’ used to make up one period of the sequence. The stagger PRI is generally used to eliminate blind speeds in MTI radar systems.

Let M denote the number of positions in one period and T is the number of periods in the pulse sequence, then stagger PRI is defined as

$$F(i + kM) = F(i) \quad i = 1, \dots, M \quad k = 1, \dots, T \tag{4.3}$$

Total number of pulses in the pulse sequence is *MT*.

Dwell and Switch PRI

The radar has bursts of pulses with several stable PRIs switched from one burst to the next. 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.

Let *M* denote the number of stages in the pulse sequence and N_i is the number of pulses in the *i*th stage, then dwell and switch PRI is defined as

First stage:

$$F(i) = F(1) \quad i = 2, \dots, N_0 \tag{4.4}$$

Other stages:

$$F(i) = F\left(1 + N_j\right) \quad i = N_j + 2, \dots, N_{j+1} \quad j = 0, \dots, M - 1 \tag{4.5}$$

Jittered PRI

Jittered PRI has large intentional PRI variations up to about 30% of the mean PRI. Such variations are generally used for ECCM purposes. It is defined by

$$F(n) = T + \varepsilon_{Gauss} \tag{4.6}$$

where *T* is the mean PRI and $\varepsilon_{\text{Gauss}}$ is a random variable which has a Gaussian distribution with zero mean and σ standard deviation.

Sliding PRI

A sliding PRI is characterized by monotonic increases or decreases in the PRI followed by a rapid switch upon reaching one extreme limit to return to the other extreme limit. This may be used to eliminate blind ranges.

Let M denote the number of pulses in one slide period, β is the minimum PRI value, and α is the slope of the modulation, then sliding PRI is defined as

$$F(n) = \alpha(n - 1 \, mod \, M) + \beta \tag{4.7}$$

Periodic PRI

Periodic PRI modulation is a nearly sinusoidal variation over a more limited range than sliding PRI. It can serve to eliminate blind ranges. It is defined by

$$F(n) = T + Asin(\omega n + \varphi) \tag{4.8}$$

where T is the mean PRI, A is the modulation amplitude (generally up to 5% of mean PRI), w is the modulation frequency (generally between 20-50 pulses per period) and φ is the phase.

Common PRI modulation types and their parameterizations are summarized in Table 4.1.

Table 4.1. Common PRI modulation types and their parameterizations

Constant	$F(n)=c \quad n=2,...,N-1$	c :a constant real number.
Stagger	$F(i+kM)=F(i)\ i=1,2,...,M$ $k=1,2,...,T$	M: the number of positions in one period T: the number of periods in the pulse sequence. Total number of pulses in the pulse sequence is $N=MT$.
Dwell and Switch	First stage: $F(i)=F(1)$ $i=2,...,N_0$ Other stages: $F(i)=F(1+N_j)$ $j=0,1,...,M-1$ $i=N_j+2,....,N_{j+1}$	M :the number of stages in the pulse sequence N_i :the number of pulses in the *ith* stage The total number of pulses in the pulse sequence is $N=\sum_{k=0}^{M-1}N_k$.
Jittered	$F(n)=T+\varepsilon_{Gauss}$	T :the mean PRI ε_{Gauss}: a random variable which has a Gaussian distribution with zero mean and σ standard deviation.
Sliding	$F(n)=\alpha(n-1 \bmod M)+\beta$	M :the number of pulses in one slide period β :the minimum PRI (Min_pri) value α = (Max_pri - Min_pri) / (M-1), the slope of the modulation.
Periodic	$F(n)=T+A\sin(wn+\varphi)$	T :the mean PRI A: the modulation amplitude (generally up to 5% of mean PRI), w: the modulation frequency (generally between 20-50 pulses per period) φ :the phase.

4.4. A Brief Introduction to Wavelets

The aim of this short section is to give some basic information about wavelets and how it is used in the multiresolution analysis of signals of interest rather than describing their properties in detail for time-frequency analysis.

4.4.1. The Haar wavelet

The Haar wavelet is defined as [37]:

$$\Psi(t) = \begin{cases} 1, & 0 \le t < \frac{1}{2} \\ -1, & \frac{1}{2} \le t < 1 \\ 0, & elsewhere \end{cases} \tag{4.9}$$

and the whole set of basis functions is obtained by dilation and translation:

$$\Psi_{m,n}(t) = 2^{-m/2}\Psi(2^{-m}t - n), \qquad m, n \in Z. \tag{4.10}$$

We call m the scale factor, since $\psi_{m,n}(t)$ is of length 2^m, while n is called the shift factor, and the shift is scale dependent ($\psi_{m,n}(t)$ is shifted by $2mn$). The normalization factor $2^{-m/2}$ makes $\psi_{m,n}(t)$ of unit norm.

A few of the basis functions are shown in Figure 4.1. It is easy to see that the set is orthonormal. At a given scale, $\psi_{m,n}(t)$ and $\psi_{m',n'}(t)$, the inner product amounts to the average of the shorter one which is zero.

$$\left\langle \Psi_{m,n}(t), \Psi_{m',n'}(t) \right\rangle = \delta[m-m']\delta[n-n'] \tag{4.11}$$

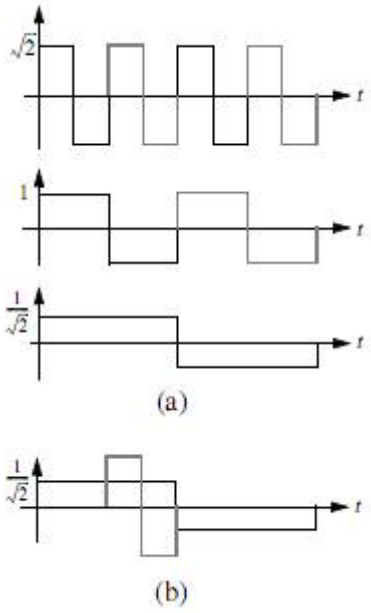

Figure 4.1. The Haar basis **(a)** A few of the Haar basis functions **(b)** Haar wavelets are orthogonal [37]

4.4.2. Discrete wavelet transform

The continuous wavelet transform (CWT) is defined by [38]:

$$W_f(a,b) = \int_{-\infty}^{\infty} \Psi_{ab}(t) f(t) dt = \langle \Psi_{ab}(t), f(t) \rangle \tag{4.12}$$

in terms of dilations and translations of a prototype or *mother* function, $\psi(t)$:

$$\Psi_{ab}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) \tag{4.13}$$

Let the m-by-n sampling lattice of (a,b) be $(a_0^m, nb_0a_0^m)$:

So that,

$$\Psi_{mn}(t) = a_0^{-m/2}\Psi\left(a_0^{-m}t - nb_0\right) \qquad m,n \in Z \tag{4.14}$$

Then the discrete wavelet transform coefficients $c_{m,n}$ are expressed as:

$$c_{m,n} = \left\langle \Psi_{mn}(t), f(t) \right\rangle = a_0^{-m/2} \int f(t)\Psi\left(a_0^{-m}t - nb_0\right)dt \tag{4.15}$$

4.4.3. Fast wavelet transform (Multiresolution decomposition algorithm)

In 1988, Mallat proposed a fast wavelet decomposition and reconstruction algorithm [39]. The Mallat's algorithm leads to a multiresolution concept based on a classical scheme known as multichannel filter banks. An easy way to construct multichannel filter banks is to cascade two-channel banks appropriately. One case can be seen in Figure 4.2, where frequency analysis is obtained by simply iterating a two-channel division on the previous low pass channel [37].

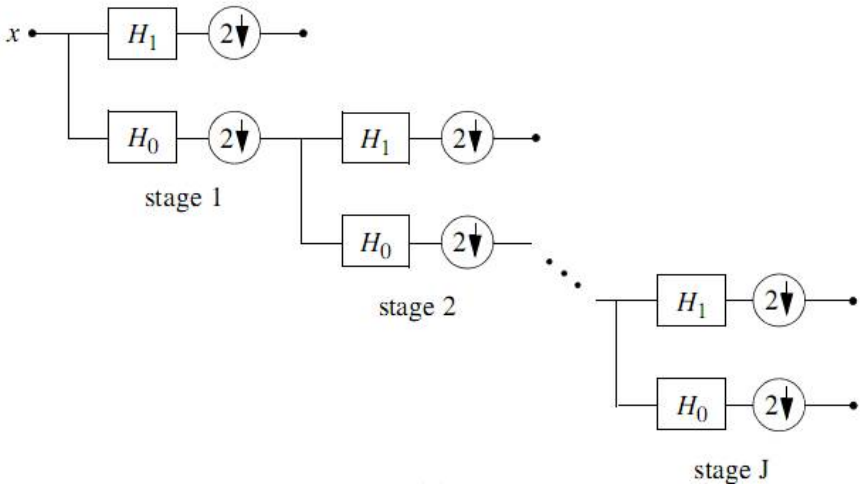

Figure 4.2. An analysis filter bank implementation. H_0 and H_1 filters correspond to lowpass and highpass filters respectively [39]

Consider the filter bank given in Figure 4.2. We see that the signal is split first via a two-channel filter bank, then the lowpass filtered part is split again using the same filter bank, and so on. At each stage, the highpass filter produces the detail (or wavelet) information, while the low pass filter produces the approximation coefficients. This structure implements a discrete-time bi-orthogonal wavelet series (we assume here that the two-channel filter banks are perfect reconstruction). If the two-channel filter bank is orthonormal, then it implements an orthonormal discrete-time wavelet series [37].

4.5. Support Vector Machines (SVM)

In this section, we will review the method of support vector machines for linearly separable binary classification problems. The material presented in this section is based on [41].

4.5.1. The optimal hyperplane algorithm

The set of labeled patterns training patterns

$$(y_1, \boldsymbol{x}_1), \dots, (y_l, \boldsymbol{x}_l), \quad y_i \in \{-1,1\} \tag{4.16}$$

is said to be linearly separable if there exists a vector $\boldsymbol{w}$ and a scalar b such that the inequalities

$$\begin{aligned} \boldsymbol{w}.\boldsymbol{x}_i + b \ &\geq 1 \quad \text{if } y_i = 1, \\ \boldsymbol{w}.\boldsymbol{x}_i + b \ &\leq -1 \quad \text{if } y_i = -1, \end{aligned} \tag{4.17}$$

are valid for all elements of the training set (4.16). We write the inequalities above in the form

$$y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) \geq 1, \quad i = 1, \ldots, l \tag{4.18}$$

The optimal hyperplane

$$\boldsymbol{w}_0.\boldsymbol{x} + b_0 = 0 \tag{4.19}$$

is the unique one which separates the training data with a maximal margin determines the direction $\boldsymbol{w}$ / $|\boldsymbol{w}|$ where the distance between the projections of the training vectors of two different classes is maximal (See Figure 4.3).

This distance $\rho\ (\boldsymbol{w}, b)$ is given

$$\rho(\boldsymbol{w}, b) = \min_{\{\boldsymbol{x}:y=1\}} \frac{\boldsymbol{x}.\boldsymbol{w}}{|\boldsymbol{w}|} - \max_{\{\boldsymbol{x}:y=-1\}} \frac{\boldsymbol{x}.\boldsymbol{w}}{|\boldsymbol{w}|} \tag{4.20}$$

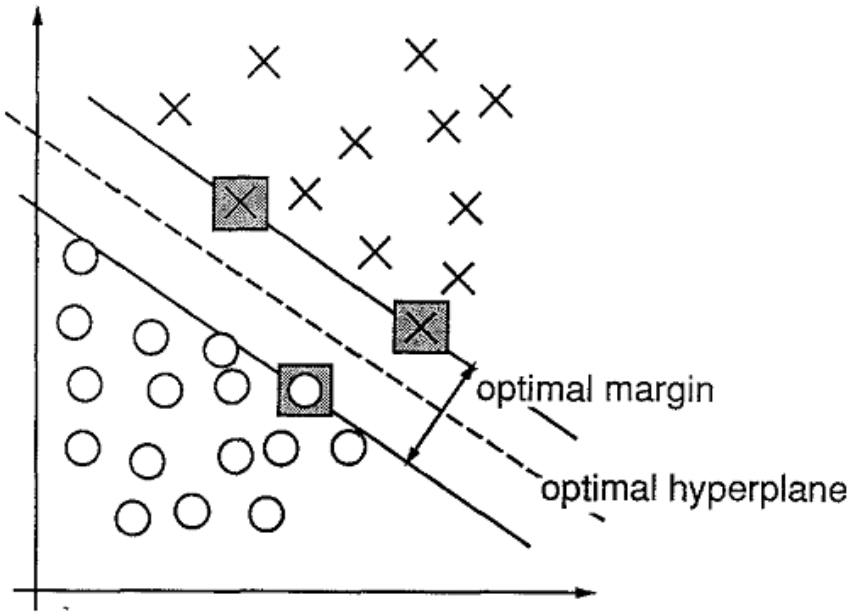

Figure 4.3. The optimal separating hyperplane and optimal margin. The support vectors, marked with grey squares, define the margin of largest separation between the classes [41]

The optimal hyperplane ($\boldsymbol{w}_0$, b_0) is the arguments that maximize the distance (4.20). It follows from (4.20) and (4.18) that

$$\rho(\boldsymbol{w}_0, b_0) = \frac{2}{|\boldsymbol{w}_0|} = \frac{2}{\sqrt{\boldsymbol{w}_0 . \boldsymbol{w}_0}} \tag{4.21}$$

This means that the optimal hyperplane is the unique one that minimizes **w .w** under the constraints (4.18). Vectors $\boldsymbol{x}_i$ for which $y_i(\boldsymbol{w}.\boldsymbol{x}_i + b)$ is equal to 1 will be termed support vectors.

4.5.2. Classification with support vector networks

Let

$$\boldsymbol{w}_0 . \boldsymbol{z} + b_0 = 0 \tag{4.22}$$

be the optimal hyperplane in feature space. The weights $\boldsymbol{w}_0$ for the optimal hyperplane in the feature space can be written as some linear combination of support vectors

$$\boldsymbol{w}_0 = \sum_{\substack{support \\ vectors}} \alpha_i \boldsymbol{z}_i \tag{4.23}$$

The linear decision function ***I(z)*** in the feature space will accordingly be of the form:

$$I(\boldsymbol{z}) = sign(\sum_{\substack{support \\ vectors}} \alpha_i \boldsymbol{z}_i \ . \boldsymbol{z} + b_0) \tag{4.24}$$

where z_i **.** z is the dot product between support vectors z_i and vector z in feature space. The decision function can therefore be described as a two layer network (Figure 4.4).

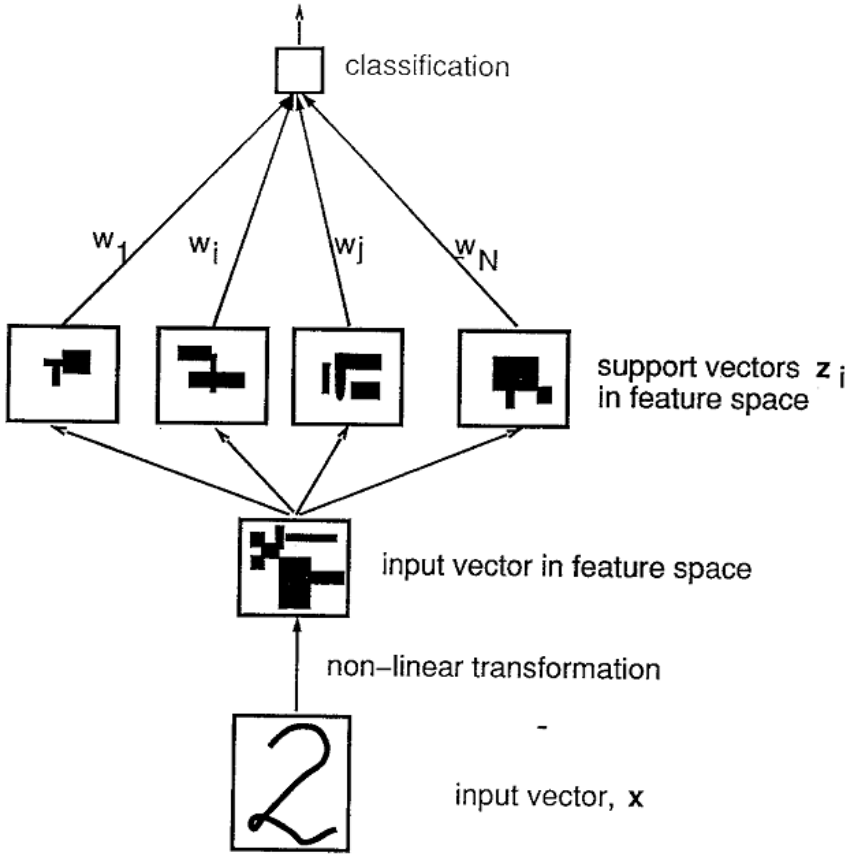

Figure 4.4. Classification by a support-vector network of an unknown pattern [41]

4.6. Higher Order Statistics

In statistics, first and second order moments, the arithmetic mean and the variance are considered to be lower order statistics. Lower order statistics describe the nature of the normal distribution. On the other hand, third and fourth order moments, the skewness and the kurtosis are the examples of higher order statistics (HOS).

These statistics describe the deviation of a distribution from the normal distribution. HOS are generally used to measure the shape parameters of

distributions (i.e., how much their shape parameters deviate from the normal distribution) and thus they are less robust than lower order statistics.

4.6.1. Skewness

Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable about its mean. The sample skewness is defined as:

$$S = E\left[\left(\frac{X - \mu}{\sigma}\right)^3\right] = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^3}{\left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{3/2}} \tag{4.25}$$

4.6.2. Kurtosis

Kurtosis is a measure of the peakedness of the probability distribution of a real-valued random variable. The sample kurtosis is defined as:

$$K = \frac{E[(X - \mu)^4]}{(E[(X - \mu)^2])^2} = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^4}{\left[\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2\right]^2} \tag{4.26}$$

4.7. Methodology

It has been shown that multiresolution signal decomposition scheme proposed in [39] can be applied to PRI estimation in intercept receivers. In doing this, variation of wavelet coefficients are closely related with PRI modulation patterns obtained from the time sequences of interleaved pulses. Classical multiresolution concept based multichannel filter banks are adopted to PRI estimation in this work.

After detailed analysis of PRI modulation types via their wavelet decompositions, it was observed that local extrema of the wavelet coefficients of jittered type modulation patterns tend to have lower magnitudes compared to stagger type modulation patterns. This is due to the fact that in staggered sequences, PRI variation is done from pulse to pulse. The radar emitter staggers from one position to another abruptly. These abrupt changes are more likely to be reflected in magnitude to their detail coefficients in contrast to jittered sequences where PRI variation is within a predefined limited range. Also, the median of the wavelet coefficients of other PRI modulation patterns (stable, dwell & switch, sliding, and periodic) tend to have lower values compared to jittered and stagger type modulation patterns. Smooth variations of those type patterns cause most of their detail coefficients tend to zero or to very low numbers in magnitude as compared to jittered and stagger type patterns. These observations gave us a chance to extract new features to distinguish between jittered, stagger and other PRI modulation types.

Radar intercept systems may encounter with a continuous stream of pulses accompanied by many imperfections, and they are required to work on a real-time basis. Since the Haar wavelet is computationally efficient and can be implemented in a transformation matrix form, it has been preferred in our study.

Feature analysis is performed on the second difference of TOAs by utilizing discrete Haar wavelet. The Haar wavelet and the set of basis functions are given in (4.9) and (4.10).

The discrete case of the wavelet can be expressed as:

$$g_{j,k}(n) = 2^{-j/2} g(2^{-j} n - k) \quad k, n \in Z, \quad j \in N. \tag{4.27}$$

where the wavelet filter *g(n)* plays the role of $\psi(t)$.

The second difference of TOAs is defined by differentiating the modulation function *F* according to Noone [29]:

$$z_n = x_{n+1} - x_n \qquad n = 1,2,..., N-2 \tag{4.28}$$

where z_n is the second difference of the time of arrival of the nth pulse. Then the detail coefficients of the wavelet decomposition of z_n at scale 2^j can be expressed as:

$$c_d(j,k) = \sum_n z_n g_{j,k}(n) \tag{4.29}$$

4.7.1. Analysis of jittered and stagger PRI modulation types

Two features are extracted from the wavelet decomposition of the vector of second differences z_n, of length M as follows:

Let E_i be the square summable energy in the ith level or the ith subband of wavelet decomposition of z_n (d^2TOA), i.e,

$$E_i = \sum_j \left\| c^i\{j\} \right\| = \sum_j \left(c^i\{j\} \right)^2 \quad j = 1,2,..., M/2^i \tag{4.30}$$

where $c^i\{j\}$ denotes the jth detail coefficient of the ith level decomposition.

The first feature is defined as a vector of the energies in L levels:

$$f_1 = \left[E_1 \; E_2 \; ... \; E_L \right] \tag{4.31}$$

where L is the effective number of decomposition levels which is analyzed in the next section.

The second feature is the magnitude of median of wavelet coefficients in the first subband:

$$f_2 = abs\left(median\left\{ c\{j\} \right\} \right) \qquad j = 1,2,..., M/2 \tag{4.32}$$

M is assumed to be a multiple of 2 which allows fast wavelet decomposition of the signal.

For classification task, we employ a cascaded form of a one dimensional binary classifier and a Support Vector Machine (SVM) classifier. SVM are from class of supervised learning algorithms that can be applied to classification or regression.

The SVM algorithm is based on the statistical learning theory developed by Vladimir Vapnik [40]. It was originally designed to solve two-class problems (binary classification), but can be easily extended to solve multi-class problems with combinations of binary classifiers. The goal of the algorithm is to determine the optimum hyperplane that separates two classes. More treatment of SVM theory is beyond the scope of this thesis and can be found in [41]. For now, it should be pointed out that major advantages of SVM are that different learning machines can be constructed by utilizing different kernels and nonlinear classification problems can be solved by linear classifiers via mapping to higher dimensional spaces without explicitly modifying the kernels [42].

We first separate the jittered and stagger modulated patterns from others by using a binary classifier. This is a one-dimensional classifier that classifies according to the absolute value of median of the wavelet coefficients in the first subband. Median of wavelet coefficients of PRI modulation types other than jittered and stagger types tend to zero or to very small values. Then, a SVM classifier is used to separate the jittered and stagger types. We use a linear kernel for evaluating the performance of our SVM classifier.

4.7.2. Analysis of other PRI modulation types

The sample kurtosis of wavelet coefficients in the *i*th subband is given by:

$$Kurt(c^i) = \frac{\frac{1}{M/2^i}\sum_{j=1}^{M/2^i}(c^i\{j\} - \mu_c{}^i)^4}{(\frac{1}{M/2^i}\sum_{j=1}^{M/2^i}(c^i\{j\} - \mu_c{}^i)^2)^2} \tag{4.33}$$

where c^i is the wavelet coefficients and μ_c^i is the sample mean of the wavelet coefficients in the *i*th subband, respectively.

Also, let the number of local extrema of wavelet coefficients in the first subband be symbolized as *local extrema* (c^l). Then the hybrid feature:

$$f_3 = [\, Kurt(c^1) \;\; Kurt(c^2) \;\; Kurt(c^3) \;\; localextrema(c^1) \,] \tag{4.34}$$

is very efficient in separating dwell & switch, sliding and sinusoidal PRI modulation types. For the sake of illustration of the separating capability of proposed feature, three kurtosis components of this feature is depicted in the simulations section.

A generalized block diagram of the proposed method is presented in Figure 4.5.

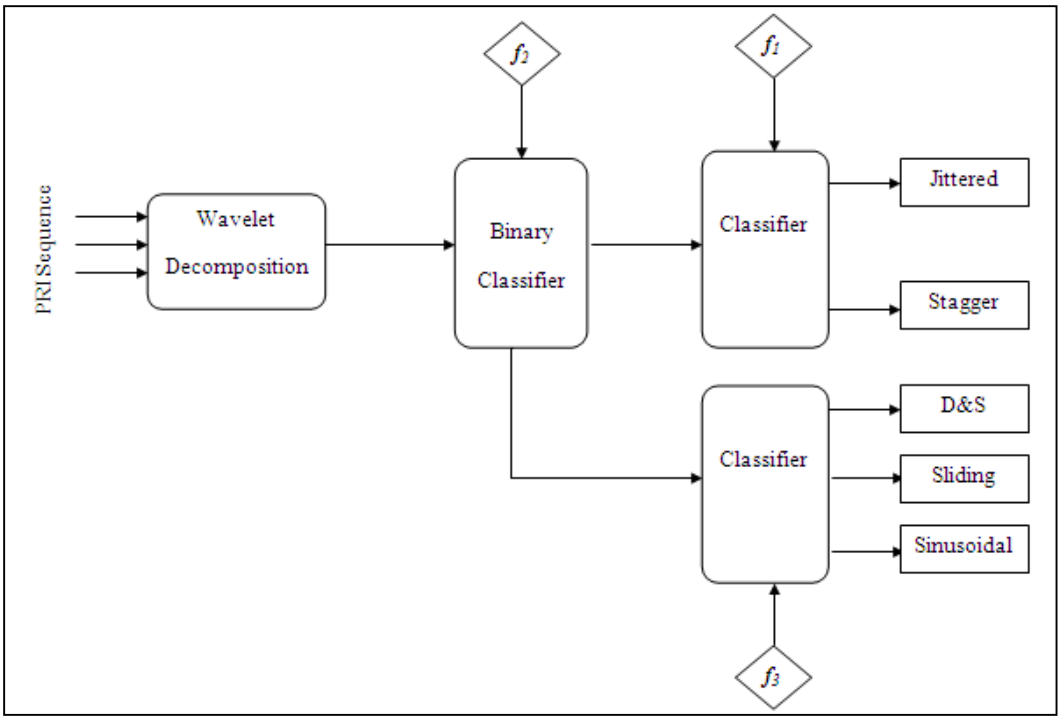

Figure 4.5. A generalized block diagram of the proposed method

4.8. Simulations

The data generation model proposed by Kauppi et al. [32] has a very high flexibility in modulation parameters and quite adequate for the generation of different PRI modulation sequences. The parameter limits used for data generation is given in Table 4.2. We train our SVM classifier for a scenario of an average missing and spurious pulses of 5%, TOA noise of 0.3% and with a very broad range of training data where limits are presented in Table 4.2.

We have created the test sequence from a broad range of PRI modulation parameters to test the separating capability of the feature set. Test sequence consists of six different type PRI sequences and their subsequences. Table 4.3 shows parameters of each PRI sequence. For each PRI sequence, subsequences are formed in such a way that they fully cover the limits of modulation parameters to ensure an unbiased sample space as possible and to see the robustness of the features against large variations of modulation parameters of PRI types.

Table 4.2. The parameter limits for synthetic data generation

PRI Modulation Types	**Parameters**	**Range**
Jittered	Jitter type Standard deviation	Gaussian, uniform 5% - 50%
Stagger	Number of positions	2 - 64
Dwell & Switch	Number of bursts Length of one burst	2 - 64 8 - 100
Sliding	Max-min ratio Number of periods	2 - 20 1 - 20
Periodic	Amplitude deviation Number of periods	4% - 50% 8 - 100
Imperfections		
Missing pulses		0% - 15%
Spurious pulses		0% - 15%
TOA uncertainty		0% - 0.4%

Table 4.3. Test sequence

PRI Sequence	Subsequences
Constant	10 subsequences.
Jittered	46 subsequences of standard deviations 5% to 50% and each subsequence has Gaussian and uniform distributions.
Stagger	63 subsequences of stagger positions 2 to 64.
Sliding	19 subsequences of max:min ratios 2 to 20 and each subsequence with periods 1,5,10,20.
Dwell-Switch	15 subsequences of number of bursts 2 to 16 and each subsequence with burst lengths 8,20, 50,100.
Periodic	47 subsequences of amplitude deviations 4% to 50% and each subsequence with periods 8, 20, 50, 100.

For the sake of illustration of separating capability of the feature set, we simulated our training data where parameter limits are presented in Table 4.2. The separating capabilities of the proposed features are illustrated in Figure 4.6 through Figure 4.8

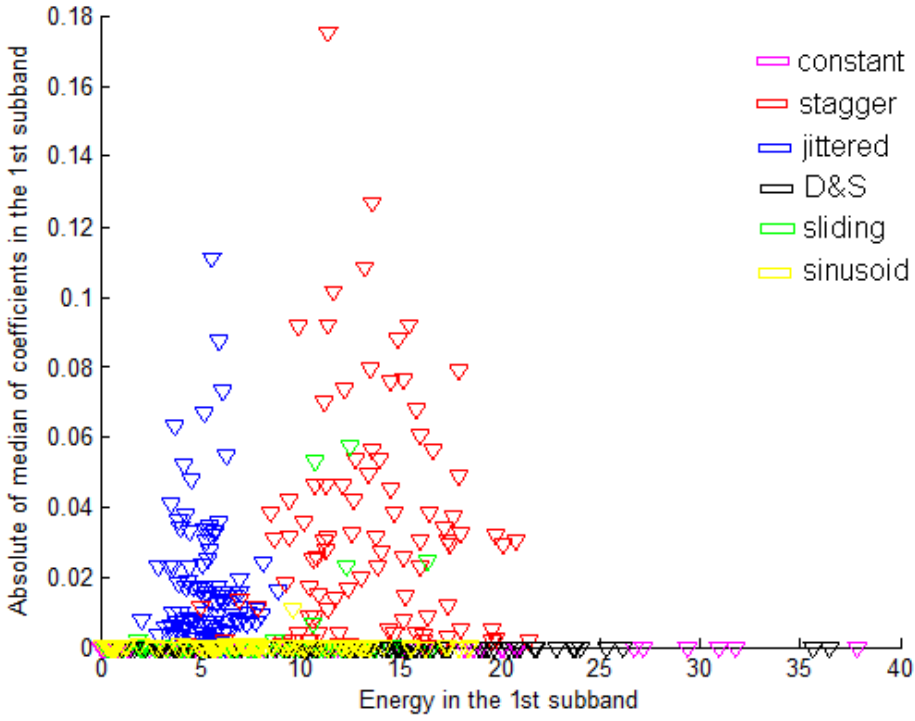

Figure 4.6. Demonstrating separating capability of the median feature

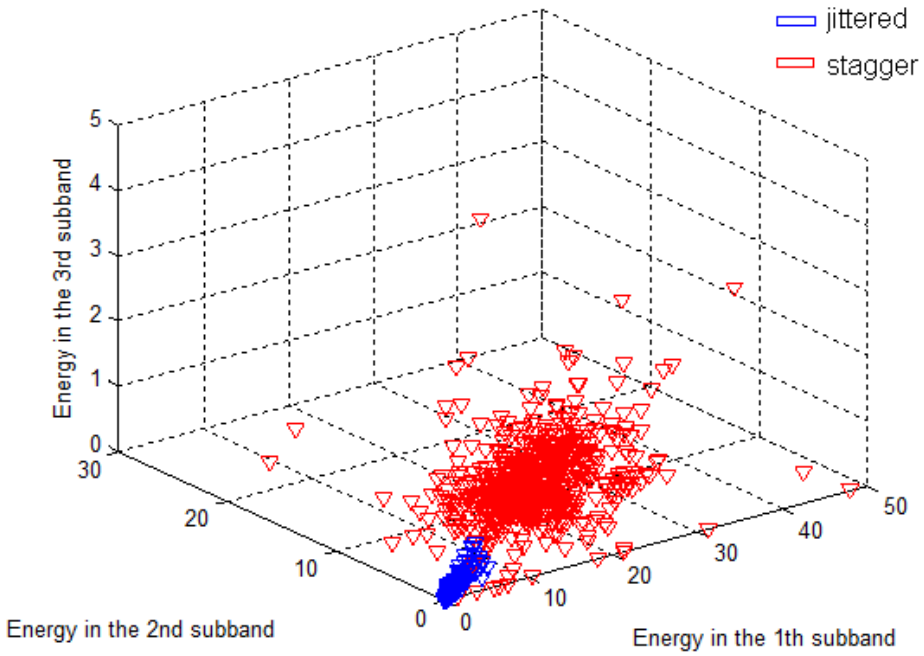

Figure 4.7. Demonstrating separating capability of the energy feature

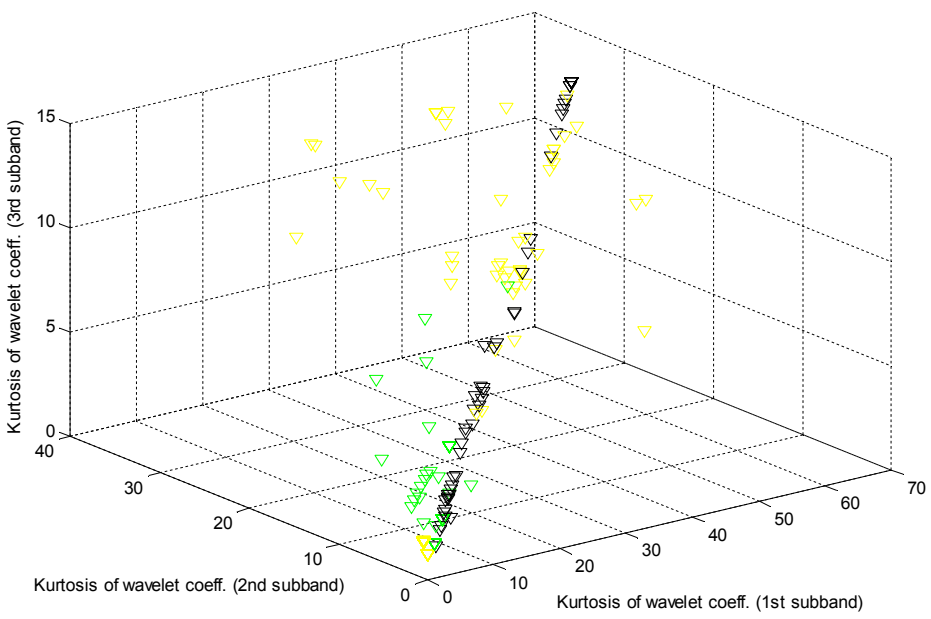

Figure 4.8. Demonstrating separating capability of three kurtosis components of the 3rd feature

Tests are performed for four distinct circumstances: in case of no imperfections, missing pulses case, spurious pulses case and in case of TOA noise. For missing and spurious pulses cases, tests are performed ten times and in case of TOA noise, the trials are increased to 100 to reflect the statistics of noise as much as possible by varying pulse repetition interval of PRI modulation types and the average recognition rate is calculated. Computation results are obtained on a standard Pentium Dual Core 2 GHz PC with MATLAB R2013a version.

The selection of L (the number of decomposition levels in the first feature) is crucial for the separating capability of the feature for jittered and stagger modulated types. A comparison about the results of Haar and Daubechies wavelets [43] for a typical scenario of an average missing and spurious pulses of 5% and TOA noise of 0.3% is given in Table 4.4. It is observed that average recognition rates of jittered and stagger sequences are very similar for both wavelets and the performance is greatly improved at two and three levels.

Table 4.4. A comparison about the results of Haar and Daubechies wavelets (M=128)

PRI Modulation Patterns	Average Recognition Rate of PRI Modulation Patterns					
	Haar wavelet			Daubechies wavelet 'db2'		
	L=1	L=2	L=3	L=1	L=2	L=3
Jittered	72	83	85	70	82	84
Stagger	99	98	98	97	98	99
Computation time (ms)	1.904	2.563	2.959	2.075	2.686	3.206

Table 4.5. Classification Results (L = 3)

PRI Modulation Patterns	AVERAGE RECOGNITION RATES OF PRI MODULATION PATTERNS (%)									
	No imperfections	Missing pulses (%)			Spurious pulses (%)			TOA noise (%)		
		5	10	15	5	10	15	0.2	0.3	0.4
Jittered	99.35	86.85	72.39	60.43	96.30	91.96	85.20	99.11	99.08	99.00
Stagger	99.21	99.52	99.37	99.21	99.53	99.50	99.36	98.95	98.90	98.22
Others	100	93	86	82	95	91	80	92	80	70

The average recognition rates of PRI modulation patterns are given in Table 4.5. It can be inferred from the results that jittered and stagger PRI modulation types have high recognition rates, usually around 95% except that recognition performance of jittered type sequences decreases rapidly as the percentage of missing pulses increases. They are not robust to missing pulses. This can be considered the only major shortcoming of the proposed method. Also, other modulation types have average recognition rates of 85%, except in the presence of TOA noise of 0.3% and 0.4%, which they are 81% and 70% respectively. Even if they decrease gradually, they still show good performance at tolerable TOA noise rates. One of the major advantages of the proposed features is that they are able to separate stagger type modulation sequences with an accuracy of around 99%. Features show great robustness to real world imperfections such as missing pulses, spurious pulses and TOA noise. This property is further analyzed in the next subsections.

4.8.1. Comparison with histogram-based methods

In [32], jittered and stagger type PRI sequences are recognized by using histogram-based features. The feature is extracted from higher order SDIF histograms and is defined as the relative strength of a stable sum in the *d*th order SDIF-histogram.

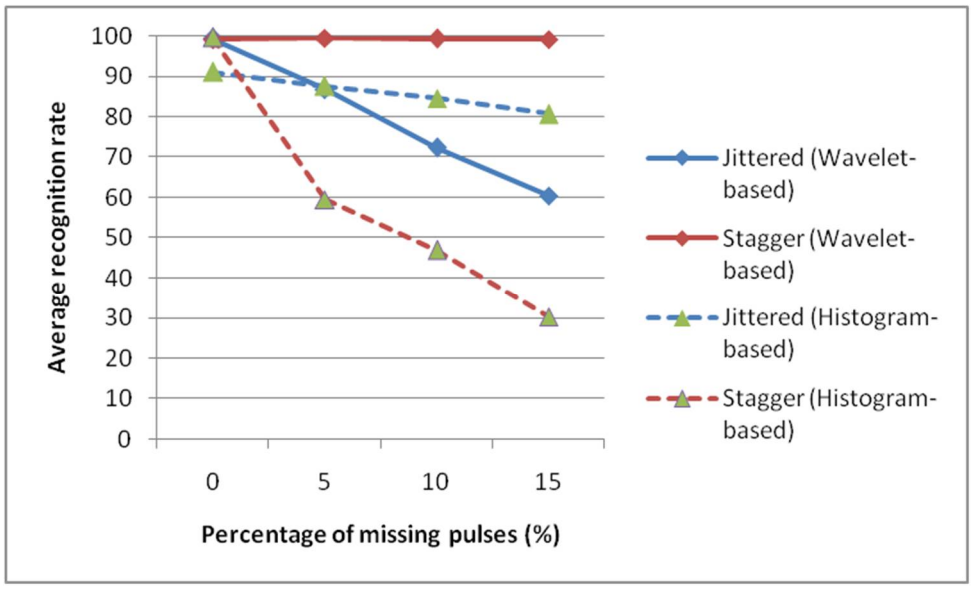

Figure 4.9. Average recognition rate of histogram and wavelet based features against missing pulses

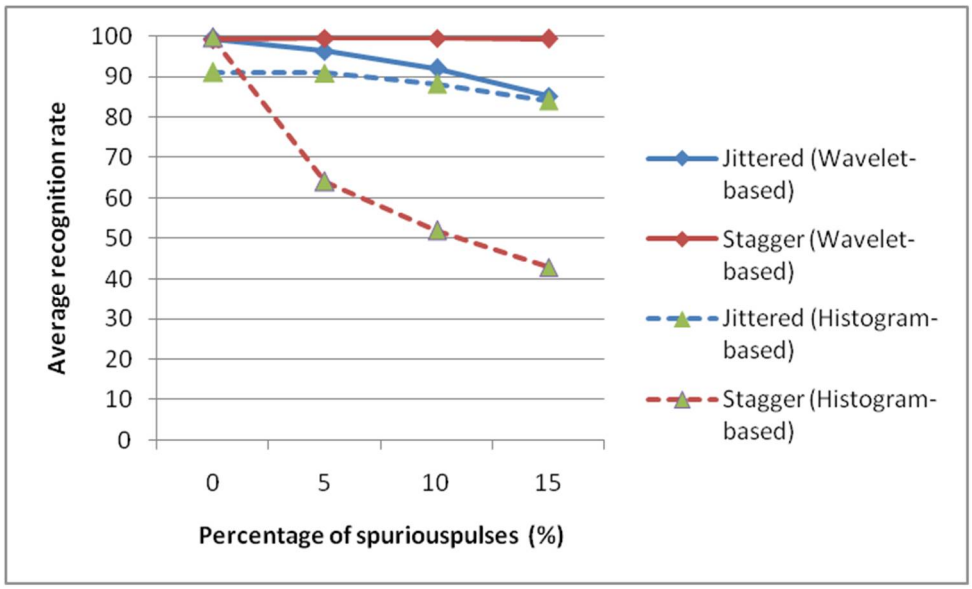

Figure 4.10. Average recognition rate of histogram and wavelet-based features against spurious pulses

The average recognition rates of jittered and stagger PRI modulation type sequences against missing and spurious pulses are given in Figure 4.9 and Figure 4.10, respectively. It is observed from Figure 4.9 that for jittered type sequences,

histogram-based features perform better than wavelet-based features. The average recognition rate of jittered sequences based on histograms is above 80%, while this rate decreases to 60% for the extreme case of 15% missing pulses when wavelet features are employed. For stagger type sequences, wavelet-based features outperform histogram-based features.

In spurious pulses case (Figure 4.10), for both jittered and stagger type sequences wavelet-based features perform better than histogram-based features. For jittered sequences, the average recognition rates of both methods is above 85%, while for stagger sequences the average recognition rate based on histograms decrease very quickly. This is due to that the dynamic range of histogram-based feature proposed in [32] for stagger type sequences rapidly increases when the number of missing or spurious pulses increases.

Histogram-based features have also some bottlenecks. First, since the number of positions of a stagger type sequence is generally unknown due to the unknown signal parameters, the feature needs be calculated up to several orders [15]. Second, the relative tolerance defining constant time interval in histogram stabilization algorithm is a choice parameter which should be updated for dynamically varying signal environments.

A comparison about the runtimes of proposed and histogram method is presented in Table 4.6. With this comparison, a superior side of the proposed method has been observed. Since the number of positions in Stagger PRI is generally unknown, histogram based features are calculated according to the highest expected number of positions in Stagger PRI [32]. A general staggered PRI sequence can contain up to 64 positions as presented in Table 4.2. On the other hand, wavelet based features do not depend on the number of positions in Stagger PRI yielding much better run-time performances as seen from the Table 4.6.

Table 4.6. A comparison about the runtime performance of proposed and histogram methods

Runtime performance proposed and histogram methods (ms) (M=128)						
Number of stagger positions	2	4	8	16	32	64
Histogram	8.686	9.250	9.544	10.535	11.764	12.970
Proposed	2.886 (does not depend on stagger positions)					

4.8.2. Robustness criteria

One of the most important contributions of this work is that the wavelet features proposed is very robust for stagger type sequences and distinguish very well as shown in Table 4.5. Figure 4.11, Figure 4.12 and Figure 4.13 show the dynamic range of the energy feature for stagger sequences with number of positions 2 to 64 against increasing number of missing pulse, spurious pulse and TOA noise percentages, respectively. It is observed that the dynamic range of the energy feature against signal imperfections does not change significantly and the calculated energy feature values vary between 10 and 25. It should be emphasized that energy feature values calculated for jittered sequences are low and this explains the high recognition performance of stagger type sequences in circumstances of missing and spurious pulses.

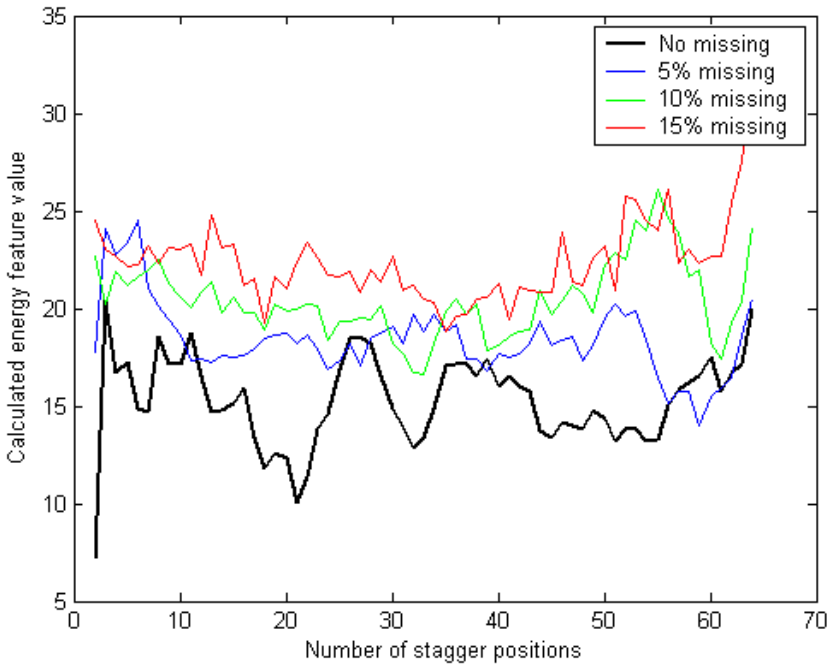

Figure 4.11. Robustness of energy feature against increasing missing pulses

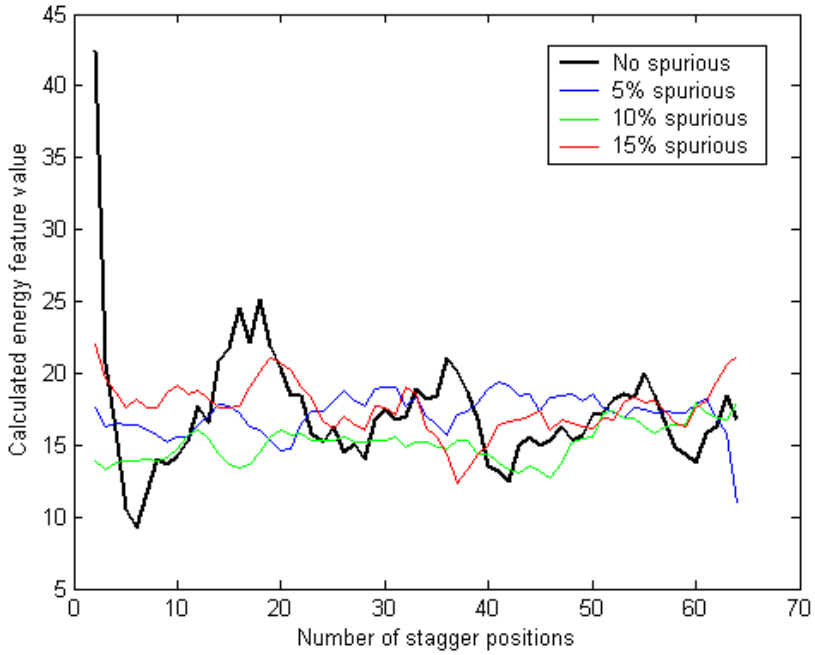

Figure 4.12. Robustness of energy feature against increasing spurious pulses

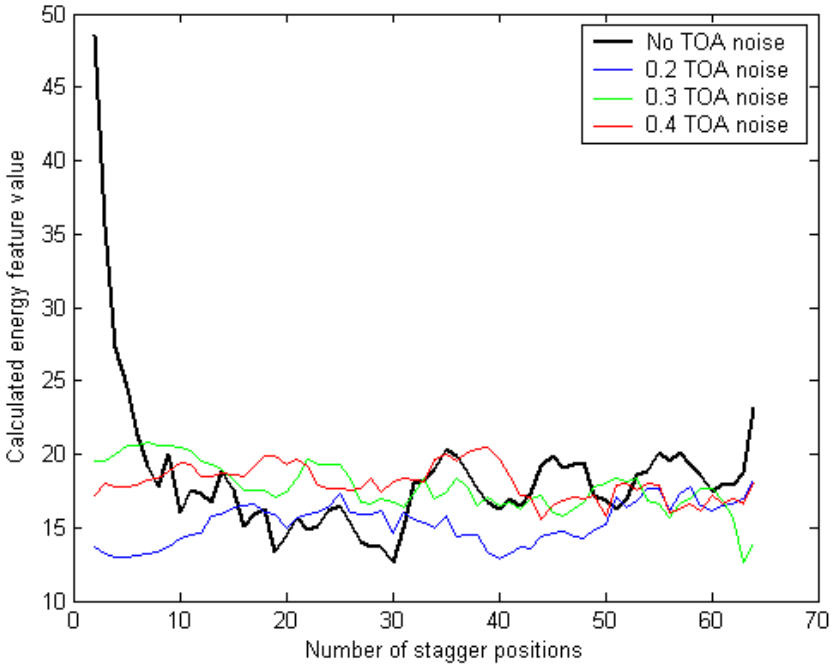

Figure 4.13. Robustness of energy feature against increasing TOA noise

It is observed from Table 4.5 that both jittered and stagger type PRI sequences are highly robust to time of arrival (TOA) noise and have average recognition rates of about 98% even in the extreme case of 0.4% noise. This is due to the fact that the wavelet features are invariant to noise. This property is depicted in Figure 4.14 and Figure 4.15 for jittered and stagger sequences, respectively. For jittered sequence, jittered sequence with jitter deviation of 20% is modeled. For stagger sequence, a stagger sequence of 4 positions is modeled.

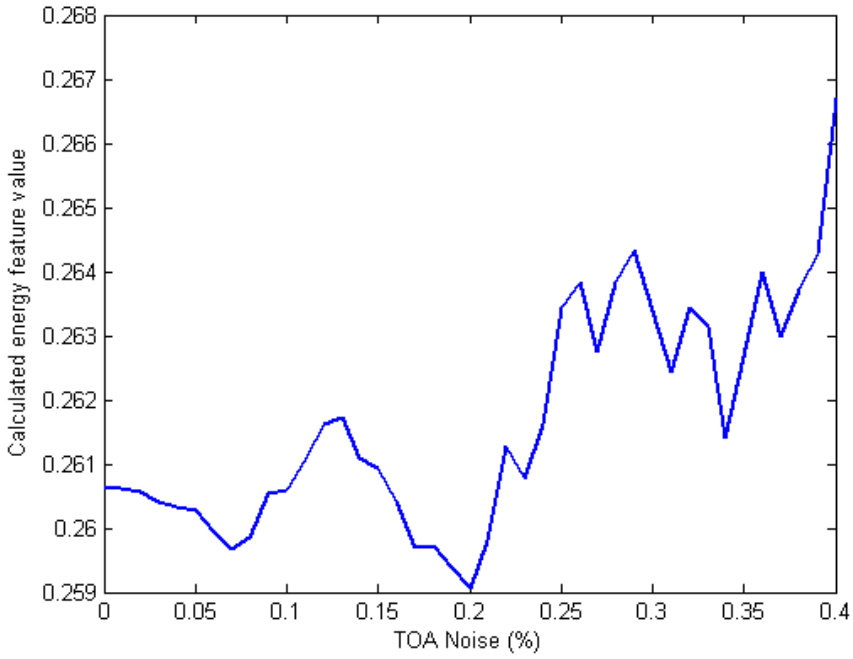

Figure 4.14. Variation of the energy feature calculated against increasing TOA noise for a jittered type PRI sequence

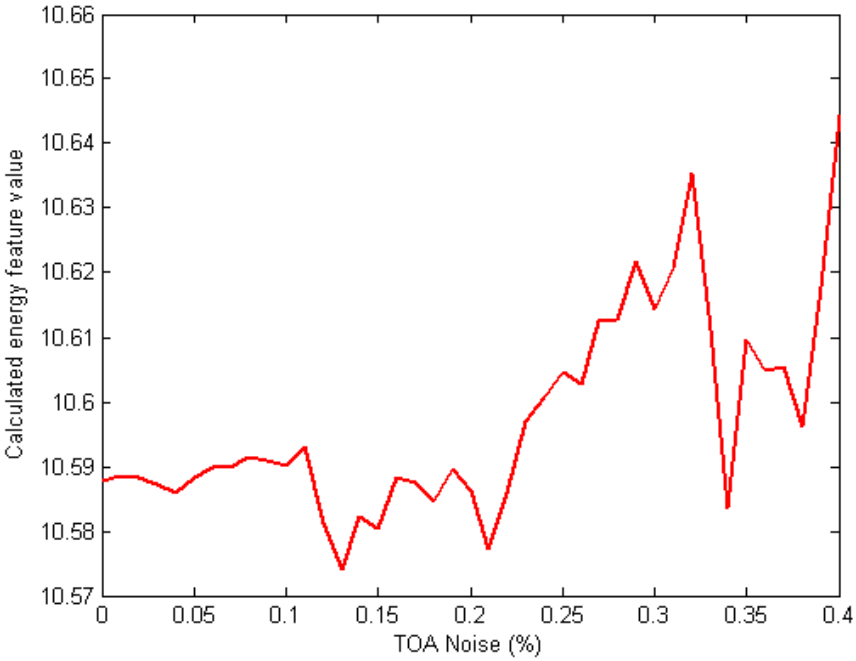

Figure 4.15. Variation of the energy feature calculated against increasing TOA noise for a stagger type PRI sequence

It can be inferred from the figures (Figure 4.14 and Figure 4.15) that for both jittered and stagger sequences, the dynamic range of the feature is nearly constant against increasing time of arrival uncertainty.

5. CONCLUDING REMARKS

In this thesis we developed new methods for radar emitter identification. The thesis is devoted to two distinct phases of radar emitter identification: pulse deinterleaving and pulse repetition interval modulation.

In Chapter 3, we developed a method for deinterleaving of radar pulse sequences in dynamically varying signal environments. For clustering task, we first evaluated the performances of two self-organizing neural networks, namely SOM and Fuzzy ART in terms of clustering accuracy, computation time and convergence. The main reason of employing self-organizing neural networks is that those types of networks lend themselves to autonomous online clustering of data which is very essential for ESM systems that have no prior knowledge about the number and categories of received pulse sequences. The results show that SOM networks can achieve better clustering scores. The sacrifice for such good clustering accuracy is relatively more computation time and longer convergence rates. On the other hand, although Fuzzy ART networks have relatively lower clustering scores, they can achieve better computation time and convergence rates. If clustering accuracy, computation time and convergence rate criteria are considered altogether, Fuzzy ART networks seem more promising algorithm for online clustering of interleaved radar data. Following the clustering task, we proposed a pulse amplitude tracking algorithm to improve deinterleaving in dynamically varying signal environments. For this purpose, we first model and derive pulse amplitude values at the receiver side. Then, we estimate the scan rate of emitters by using Gauss-Newton algorithm. It is shown that the proposed algorithm improves deinterleaving of interleaved radar pulses and is promising for such electronic warfare environments.

In Chapter 4, we developed a method for recognizing pulse repetition interval modulation patterns. The method incorporates features based on the multiresolution wavelet analysis of different types of pulse repetition interval modulated sequences. Three wavelet features were found to be distinctive to separate jittered, stagger and other modulation patterns. For classification task,

we employ a cascaded form of one-dimensional binary classifier and SVM classifier. We test our method for a broad range of PRI modulation parameters. Simulation results show that jittered and stagger PRI modulation patterns have high recognition rates in circumstances of missing pulses, spurious pulses, and TOA noise. One of the most important contributions of this thesis is that the proposed wavelet features is very robust for stagger type sequences and outperform conventional histogram based methods. Besides, the computation time of wavelet based features does not depend on stagger positions and yields much better run-time performance compared to histogram-based features which need to be calculated according to the highest expected number of positions in Stagger PRI.

In future work, real time implementation of the proposed methods on hardware is aimed.

APPENDIX

A.1. SCENARIO-BASED SYNTHETIC MIXED PULSE GENERATOR

Modeling environments consisting of multiple emitters is important for realistic implementation of an ESM receiver that integrates radar pulses and processes them. For this reason, a so-called synthetic mixed pulse generator (simulator) is designed.

The simulator is capable of creating emitters with specified parameters of PRI type, number of pulses, frequency, pulse width, angle of arrival, scan rate (rotation period), and given distance.

The simulator supports various PRI Types including stable, stagger, dwell& switch, jittered, sliding and sinusoidal types.

The simulator generates a Pulse Description Word (PDW) for each pulse. A PDW consists of Time of Arrival (TOA), pulse frequency, pulse width, pulse amplitude and angle of arrival (AOA) parameters. A measurement noise of Gaussian type with zero mean and given variance is added to all parameters. Then, it combines PDWs to form an Emitter Description Word (EDW).

The simulator interleaves all pulses in a natural time order and sorts them according to their TOA values. To implement electronic timing noise (jitter), which has a Gaussian distribution, a zero mean, independent Gaussian noise of given variance is added to all TOA values.

The simulator can also simulate a rotating radar antenna. It derives pulse amplitudes from a given antenna pattern, scan rate of the antenna (or rotation period), and distance of the emitter to the ESM receiver. Distance is used to calculate path loss for each emitter at specific frequencies. A zero mean, independent Gaussian noise of given variance is added to pulse amplitude values to implement measurement errors.

The simulator also implements situations in which some of pulses are missing from the pulse sequence or some spurious pulses are interfering to the pulse sequence. In missing pulses case, a predetermined percent of pulses are dropped from the pulse sequence randomly and a new Mixed Emitter Description Word (EDW) is formed. In spurious pulses case, a predetermined percent of random pulses (that may originate from other objects) are mixed with the pulse sequence and a new Mixed Emitter Description Word (EDW) is formed.

Some screen snapshots of the designed simulator are presented in Figure A.1 and Figure A.2.

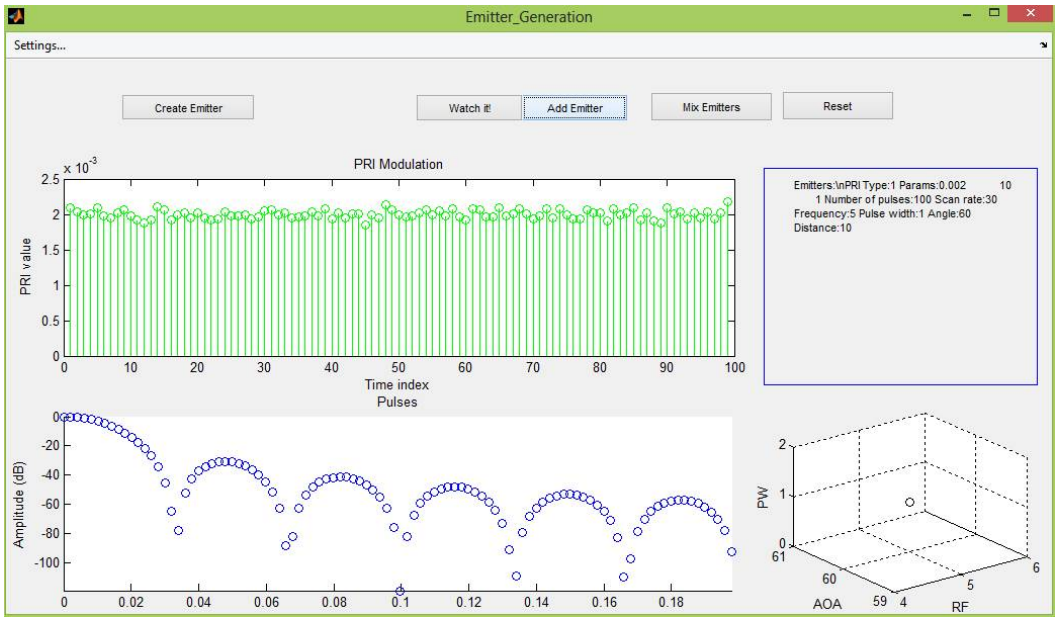

Figure A.1. Main GUI of the synthetic mixed pulse generator

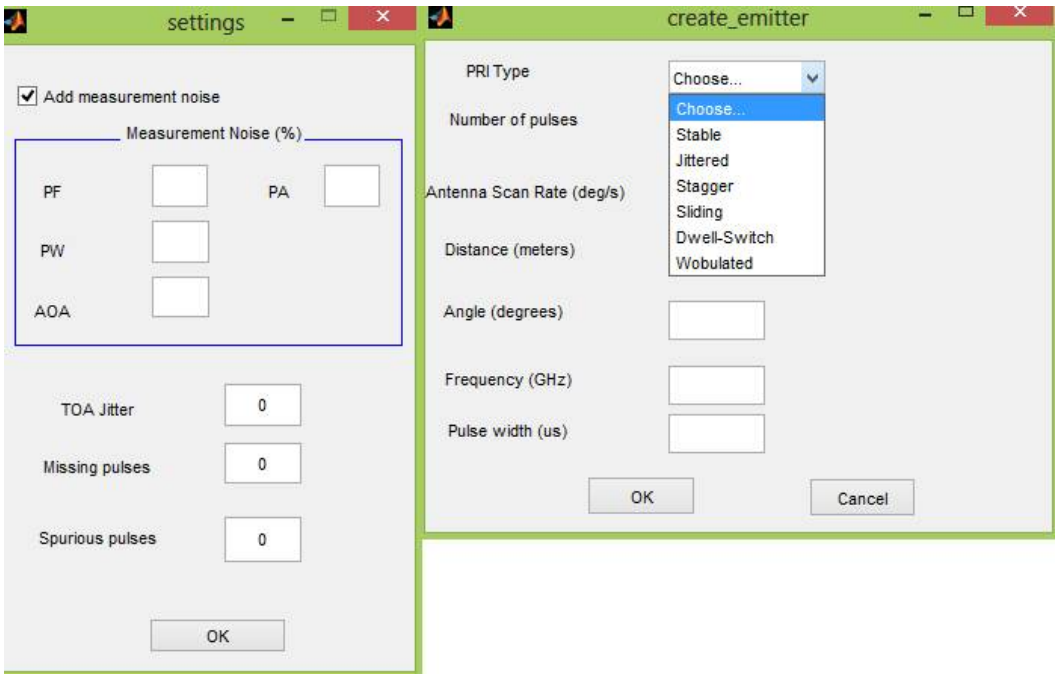

Figure A.2. Settings and emitter creation submenu

Simulation Examples

Two examples are presented to show the results of simulation of two radars and the mixed signal obtained by the ESM receiver.

Example 1:

Radar Type: Navigation and Search

Model: APS-705

Frequency band: 8-10 GHz

Pulse width: 0.05 and 1.5 μs

PRF: 1,600 and 650 Hz

Antenna Rotation Period: 20 or 40 rpm (scan rate: 120 deg/s or 240 deg/s)

PRI modulation pattern, scan pattern (Amplitude vs. TOA plot) and three dimensional (RF, AOA, PW) plot of emitter characteristics regarding to this radar are illustrated in Figure A.3, Figure A.4 and Figure A.5, respectively.

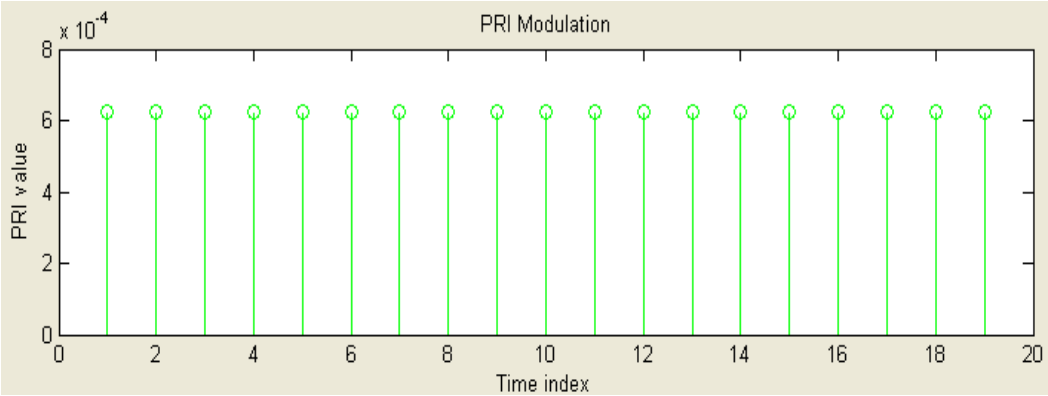

Figure A.3. PRI Modulation pattern

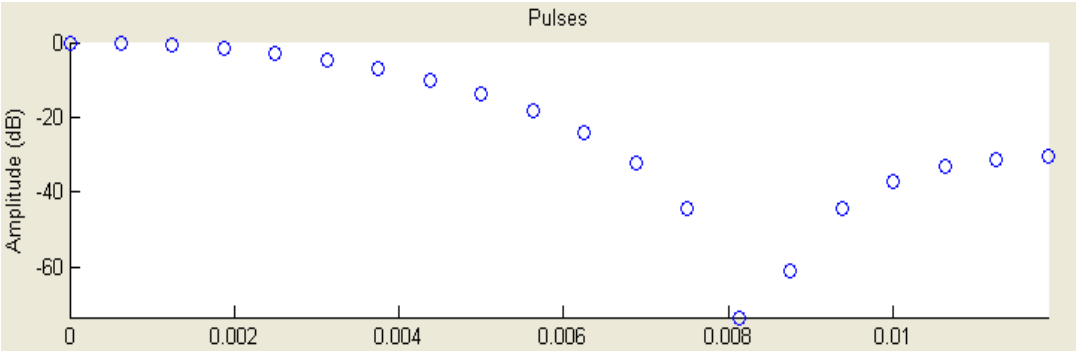

Figure A.4. Scan pattern (Amplitude vs. TOA plot)

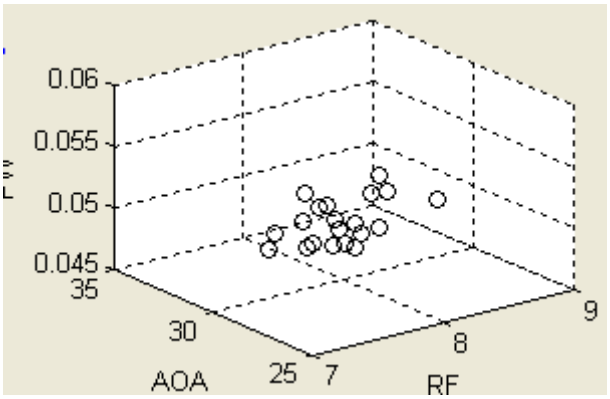

Figure A.5. Three dimensional plot of PF, AOA, and PW

Example 2:

Radar Type: Surveillance

Model: AN/APS-128/-128

Frequency band: 9,375 MHz

Pulse width: 2.4 and 0.5 µs

PRF: 400, 1200, 1600 Hz (stagger)

Antenna Rotation Period: 15 rpm (scan rate: 90 deg/s)

PRI modulation pattern, scan pattern (Amplitude vs. TOA plot) and three dimensional (RF, AOA, PW) plot of emitter characteristics regarding to this radar are illustrated in Figure A.6, Figure A.7 and Figure A.8, respectively.

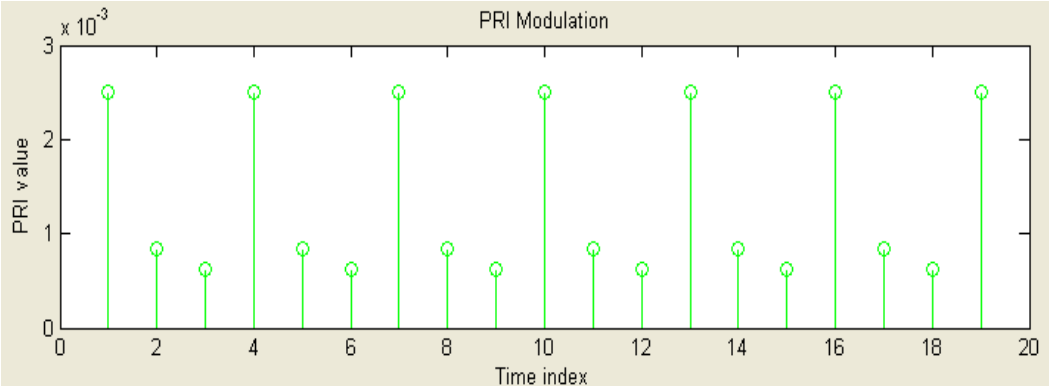

Figure A.6. PRI Modulation

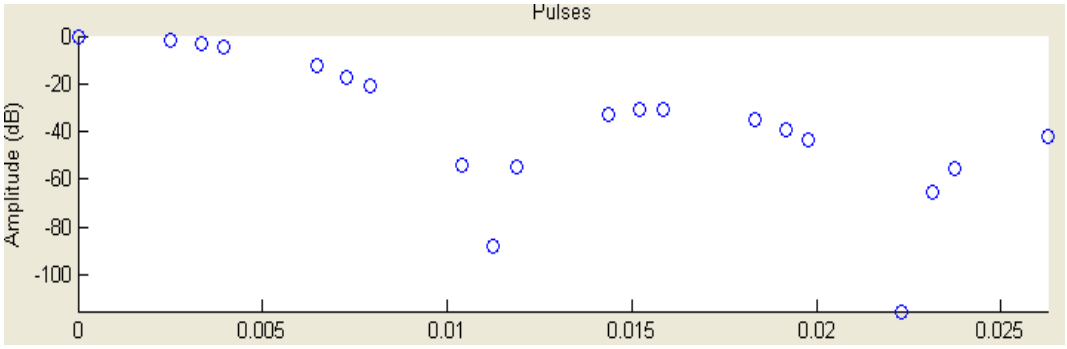

Figure A.7. Amplitude vs. TOA plot

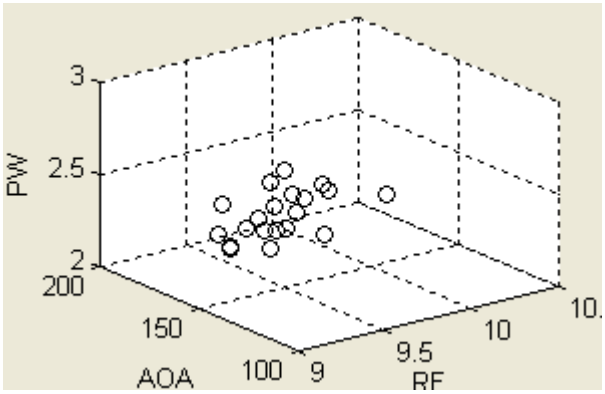

Figure A.8. Three dimensional plot of PF, AOA, and PW

The scan pattern of the mixed emitters observed by the ESM receiver and the resulting three-dimensional plot of mixed pulse sequence (RF, AOA, PW) are presented in Figure A.9 and Figure A.10, respectively.

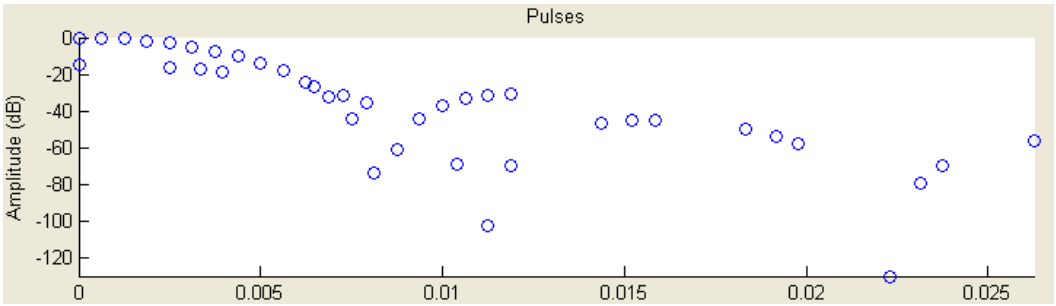

Figure A.9. Amplitude vs. TOA plot of mixed emitters

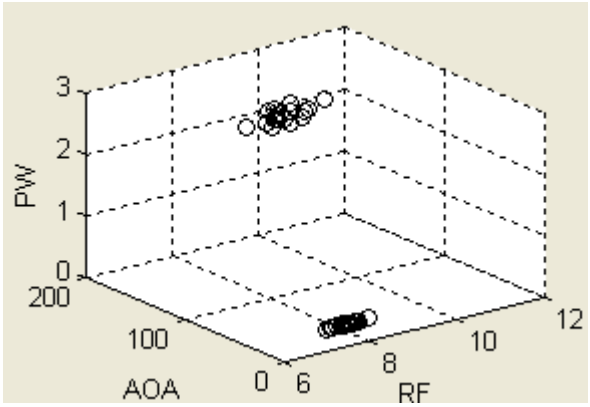

Figure A.10. Three dimensional plot (PF, AOA, and PW) of mixed emitters

A.2. MATLAB SCRIPTS OF COMMON PRI MODULATION TYPES

This appendix presents the MATLAB code scripts of common PRI modulation types for quick reference.

Constant_PRI_Generate.m

```
function pulse_toa_values = Constant_PRI_Generate(number_of_pulses, pri_s)

% This function generates a constant (stable) type PRI % modulation sequence.
% Inputs:
% number_of_pulses - Number of pulses to be generated
% pri_s - stable PRI value
% Output: pulse_toa_values - PRI sequence

pulse_toa_values = [];
pulse_toa_values(1) = 0;

for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s;
end
```

DwellSwitch_PRI_Generate.m

```
function pulse_toa_values = DwellSwitch_PRI_Generate(number_of_pulses,
no_bursts, len_burst)

% This function generates a dwell&switch type PRI
% modulation sequence.
% Inputs:
% number_of_pulses - Number of pulses to be generated
% no_bursts - number of bursts: between 2-16
% len_burst - length of one burst: between 8-100
% Output: pulse_toa_values - PRI sequence

j=1;
for k=1:no_bursts
   burst_pri = rand(1);
for count=1:len_burst
      pri_s(j) = burst_pri;
      j = j+1;
end
end
```

```
pulse_toa_values = [];
pulse_toa_values(1) = 0;

pri_index = 1;
for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s(pri_index);
   pri_index = pri_index + 1;
if pri_index > length(pri_s)
      pri_index = 1;
end
end
```

Jittered_PRI_Generate.m

```
function pulse_toa_values =
Jittered_PRI_Generate(number_of_pulses,mean_pri_jitter,deviation_pri_jitter,
distr_type_jitter)

% This function generates a jittered type PRI
% modulation sequence.
% Inputs:
% number_of_pulses - Number of pulses to be generated
% mean_pri_jitter -  mean PRI
% deviation_pri_jitter - percentage of PRI deviation:   between 5-50%
% distr_type_jitter - distribution type 1:gaussian
2:uniform 3:discrete
% Output: pulse_toa_values - PRI sequence

if distr_type_jitter == 1 %gaussian
   std_dev_pri_jitter = (mean_pri_jitter*deviation_pri_jitter/100) / 3;
% 6-sigma rule. 99.7 percent of samples.
   pri_s = normrnd(mean_pri_jitter, std_dev_pri_jitter, 1, number_of_pulses);

elseif distr_type_jitter == 2 %uniform
   min_pri_jitter = mean_pri_jitter*(1-deviation_pri_jitter/100);
   max_pri_jitter = mean_pri_jitter*(1+deviation_pri_jitter/100);
   pri_s = unifrnd(min_pri_jitter, max_pri_jitter, 1, number_of_pulses);

elseif distr_type_jitter == 3 %discrete
   % number of discrete pris: between 32-64
   no_dicrete_pris = round(32 + (64-32) * rand(1));
   discrete_pri_set = rand(1, no_dicrete_pris);
for k=1:number_of_pulses
      % choose from discrete pri   set
      pri_s(k) = discrete_pri_set(ceil( no_dicrete_pris*rand(1) ));
end
```

```
end

pulse_toa_values = [];
pulse_toa_values(1) = 0;

for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s(i);
end
```

Stagger_PRI_Generate.m

```
function pulse_toa_values = Stagger_PRI_Generate(number_of_pulses,
no_stages, stagger_pris)

% This function generates a staggered type PRI
% modulation sequence.
% Inputs:
% number_of_pulses - Number of pulses to be generated
% no_stages - number of stagger positions in one period: between 2-64
% stagger_pris - set of stagger PRI values (positions)
% Output: pulse_toa_values - PRI sequence

% number of periods: between 5-160 (RESERVED)
for k=1:no_stages
   pri_s(k) = stagger_pris(k);
end

pulse_toa_values = [];
pulse_toa_values(1) = 0;

pri_index = 1;
for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s(pri_index);
   pri_index = pri_index + 1;
if pri_index > length(pri_s)
      pri_index = 1;
end
end
```

Sliding_PRI_Generate.m

```
function pulse_toa_values = Sliding_PRI_Generate(number_of_pulses,
max_min_ratio, max_sliding_pri, no_periods, sliding_type)

% This function generates a sliding type PRI
% modulation sequence.
```

```
% Inputs:
% number_of_pulses - Number of pulses to be generated
% max_min_ratio - deviation ratio: max/min between 2:20
% max_sliding_pri - maximum PRI
% no_periods - number of periods: between 1-20
% sliding_type- Sliding type 1:increasing 2:decreasing
% Output: pulse_toa_values - PRI sequence

% minimum PRI
min_sliding_pri = max_sliding_pri / max_min_ratio;
% number of sliding pri values in each period
number_of_pri_values = round(number_of_pulses / no_periods);

if sliding_type == 1
   pri_s(1) = min_sliding_pri;
for i=2:number_of_pri_values-1
      pri_s(i) = min_sliding_pri + (i-1)*(max_sliding_pri -
min_sliding_pri)/(number_of_pri_values-1);
end
   pri_s(number_of_pri_values) = max_sliding_pri;
end

if sliding_type == 2
   pri_s(1) = max_sliding_pri;
for i=2:number_of_pri_values-1
      pri_s(i) = max_sliding_pri - (i-1)*(max_sliding_pri -
min_sliding_pri)/(number_of_pri_values-1);
end
   pri_s(number_of_pri_values) = min_sliding_pri;
end


pulse_toa_values = [];
pulse_toa_values(1) = 0;

pri_index = 1;
for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s(pri_index);
   pri_index = pri_index + 1;
if pri_index > length(pri_s)
      pri_index = 1;
end
end
```

Periodic_PRI_Generate.m

```
function pulse_toa_values = Periodic_PRI_Generate(number_of_pulses,
mean_pri_periodic, periodic_ampl, no_periods)

% This function generates a periodic type PRI
% modulation sequence.
% Inputs:
% number_of_pulses - Number of pulses to be generated
% mean_pri_periodic - mean PRI value
% no_periods - number of periods: between 1-20
% periodic_ampl - deviation from mean pri: between 4-50%
% Output: pulse_toa_values - PRI sequence

% frequency
periodic_freq = round(number_of_pulses / no_periods);

for i=0:periodic_freq-1
   pri_s(i+1) = mean_pri_periodic * (1 +
periodic_ampl/100*sin(2*pi*i/periodic_freq));
end


pulse_toa_values = [];
pulse_toa_values(1) = 0;

pri_index = 1;
for i=1:number_of_pulses-1
   pulse_toa_values(i+1) = pulse_toa_values(i) + pri_s(pri_index);
   pri_index = pri_index + 1;
if pri_index > length(pri_s)
      pri_index = 1;
end
end
```

REFERENCES

[1] Aslan M.K., *Emitter Identification Techniques in Electronic Warfare*, M. S. Thesis in Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, Dec., 2006.

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

[3] Adamy, D.L., *Introduction to Electronic Warfare: Modeling and Simulation*, Artech House, Inc., 2002.

[4] Olgun M.E., *Design and FPGA Implementation of an Efficient Deinterleaving Algorithm*, M. S. Thesis in Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, Aug., 2008.

[5] Güven E., *Emitter Identification in Electronic Warfare by the Use of Clustering Techniques*, M. S. Thesis in Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, Sep., 1994.

[6] Güvenlik A.R., *Clustering Techniques for Emitter Identification in Electronic Warfare*, M. S. Thesis in Electrical and Electronics Engineering, Middle East Technical University, Ankara, Turkey, Dec., 1999.

[7] Wiley, R. G., ELINT*: The Interception and Analysis of Radar Signals*, Artech House, Inc., 2006.

[8] Kauppi, J.-P., Martikainen, K., "An efficient set of features for pulse repetition interval modulation recognition", *International Conference on Radar Systems*, pp. 1-5, 2007.

[9] Filippo, N., *Introduction to Electronic Defence Systems, 2nd Edit.*, Artech House, Inc., 2001.

[10] Adamy, D.L., *A Second Course in Electronic Warfare*, Artech House, Inc., 2004.

[11] Skolnik, M.I., *Radar Handbook, 3rd Edit.*, McGraw-Hill International Edition, 2008.

[12] Orfanidis S.J., *Electromagnetic Waves and Antennas*, Rutgers University (available online), 2002.

[13] Mardia H. K., "Adaptive Multi-dimensional Clustering for ESM", *IEE Colloquium on Signal Processing for ESM Systems*, 1988.

[14] Mardia, H.K., "New techniques for the deinterleaving of repetitive sequences", *IEE Proc. F, Commun. Radar & Signal Process.*, 136, (4), pp. 149-154, 1989.

[15] Milojevic, D.J. and Popovic, B.M., "Improved Algorithm for Deinterleaving of Radar Pulses," *IEE Proc. F., Comm., Radar & Signal Processing*, vol. 139, no. 1, pp. 98-104, 1992.

[16] Nishiguchi, K. and Kobayashi, M. "Improved algorithm for estimating pulse repetition intervals", *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, pp.407 - 421 , 2000.

[17] Moore, J.B. and Krishnamurthy, V., "Deinterleaving of Pulse Trains Using Discrete-Time Stochastic Dynamic Linear Models," *IEEE Trans. on Signal Processing*, vol. 42, no. 11, Nov., 1994.

[18] Shieh, C.-S. and Lin C.-T., "A vector neural network for emitter identification", *IEEE Trans. Antennas Propag.*, vol. 50, no. 8, pp.1120 - 1127 , 2002.

[19] Orsi R.J., Moore J.B., and Mahony R.E., "Spectrum estimation of interleaved pulse trains," *IEEE Trans. on Signal Processing*, vol. 47, no. 6, pp. 1646–1653, June 1999.

[20] Driscoll, D. E., Howard, S. D., "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, 1999.

[21] Ray P.S., "A novel pulse TOA analysis technique for radar identification", *IEEE Transactions on Aerospace and Electronic Systems* 34 (3), 1998.

[22] Ata'a A.W., Abdullah S.N., "Deinterleaving of radar signals and PRF identification algorithms", *IET Radar Sonar Navig*., 1, (5), pp. 340–347, 2007.

[23] Liu J., Lee J.P.Y., Li L., Luo Z-Q., Wong K.M., "Online clustering algorithms for radar emitter classification", *IEEE Trans. on Neural Networks.*, vol. 27(8), pp.1185 - 1196 , 2005.

[24] Granger, E., Savaria, Y., Lavoie, P., and Cantin, M., "A comparison of self-organizing neural networks for fast clustering of radar pulses", *Signal Process.*, 64, (3), pp. 249–269, 1998.

[25] Vesanto J., Alhoniemi Esa, "Clustering of the Self-Organizing Map", *IEEE Trans. on Neural Networks.*, vol. 11(3), pp.586 - 600 , 2000.

[26] T. Kohonen, *Self-Organizing Maps*. Berlin/Heidelberg, Germany: Springer, vol. 30, 1995.

[27] Carpenter G.A., Grossberg S., and Rosen D.B., "Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system", *Neural Networks.*, vol. 4, pp.759 - 771, 1991.

[28] Zadeh L.A., "Fuzzy Sets", *Information and Control*, 8, 338-353, 1965.

[29] Noone G.P., "A neural approach to automatic pulse repetition interval modulation recognition", *Information, Decision and Control Conference (IDC '99)*, Adelaide, South Australia, USA: IEEE pp. 213–218, 08-10 Feb 1999.

[30] Rong H., Jin W., Zhang C., "Application of support vector machines to pulse repetition interval modulation recognition", *2006 6th international conference on ITS telecommunications*, Chengdu, China, USA: IEEE. pp. 1187–1190, June 2006.

[31] Ryoo Y.J., Song K.H., Kim W.W., "Recognition of PRI modulation types of radar signals using the autocorrelation", *IEICE T Commun*, vol. 90 pp. 1290–1294, 2007.

[32] Kauppi J.P, Martikainen K., Ruotsalainen U., "Hierarchical classification of dynamically varying radar pulse repetition interval modulation patterns", *Neural Networks* vol. 23 pp. 1226-1237, 2010.

[33] Hu G., Liu Y., "An Efficient method of pulse repetition interval modulation recognition", *2010 International Conference on Communications and Mobile Computing (CMC)*, Shenzhen, China, USA: IEEE. pp. 287–291, 12-14 April 2010.

[34] Mahdavi A., Pezeshk A.M., "A Robust method for PRI modulation recognition", *2010 IEEE 10th International Conference on Signal Processing (ICSP)*, Beijing, China, USA: IEEE. pp. 1873-1876, 24-28 Oct. 2010.

[35] Song K.H, Lee D.W, Han J.W, Park B.K., "Pulse repetition interval modulation recognition using symbolization", *2010 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, Sydney, Australia, USA: IEEE. pp. 540-5451, 3 Dec. 2010.

[36] Keshavarzi M., Pezeshk A.M., Farzaneh F., "A new method for detection of complex Pulse Repetition Interval Modulations", *2012 IEEE 11th International Conference on Signal Processing (ICSP)*, Beijing, China, USA: IEEE. pp. 1705-1709, 21-25 Oct. 2012.

[37] Vetterli M., Kovacevic J., *Wavelets and Subband Coding*, Prentice Hall, Englewood Cliffs, NJ, USA, 2007.

[38] Akansu A.N. and Haddad R.A., *Multiresolution Signal Decomposition, 2nd Edition*, Academic Press, London, U.K., 2001.

[39] Mallat S. "A theory for multiresolution signal decomposition: the wavelet representation", *IEEE Trans. on Pattern Analysis*, vol. 11, pp. 674-693, 1989.

[40] Vapnik V.N., "An overview of statistical learning theory", *IEEE Trans. on Neural Networks*, vol 10, pp. 988–1000, 1999.

[41] Cortes C., Vapnik V., "Support-Vector Networks", *Machine Learning*, vol. 20, pp. 273–297, 1995.

[42] Müller K.R, Mika S., Rätsch G., Tsuda K., Schölkopf B., "An introduction to kernel-based learning algorithms", *IEEE Trans. on Neural Networks*, vol. 12, pp. 181-201, 2001.

[43] Daubechies I. *Ten lectures on wavelets*. SIAM, Philadelphia, USA, 1992.

[44] Gençol K., At N., Kara A., "A wavelet based feature set for pulse repetition interval modulation recognition", *Turkish Journal of Electrical Engineering and Computer Sciences*, (accepted for publication).