# Deinterleaving of radar signals and PRF identification algorithms

A.W. Ata'a and S.N. Abdullah

**Abstract:** Electronic warfare (EW) receivers are passive receivers which receive emissions from other platforms, and do certain analysis on these emissions. Some EW receivers receive radar pulses, measure the parameter of each pulse received and group the pulses that belongs to the same emitter together to determine the radar parameters for each emitter. These parameters are then compared with values stored for known radar types, to identify the emitter type. Two parts are focused, emitters deinterleaving and PRF-type identification. The deinterleaving is done through parameters clustering. Two parameters are selected for clustering direction of arrival and radio frequency. A self-organising neural network called Fuzzy ART is proposed for clustering. This algorithm has a very good clustering quality and can run in real-time applications.The PRF-type identification is done through time-of-arrival (TOA) analysis. Three previously presented algorithms are combined in new scheme to do the TOA analysis (or PRF-type identification). These algorithms are difference TOA histogram, TOA folding histogram and sequence search algorithm. The complete proposed system has been tested using three different tests. These tests are simple PRI test, jittered PRI test and staggered PRI test. The proposed system identifies up to 90 simple emitters, 20 jittered emitters and 20 staggered emitters. In all tests, the data were simulated and generated using software.

## 1 Introduction

Once radar was introduced in World War II, ways to jam and defeat its effectiveness were under investigation. In order to effectively interfere with a radar, one must at least know whether there is a radar target. Therefore an electronic warfare (EW) receiver is needed to detect the existence of hostile radar signals.

Now days, in the more probable scenarios, the environment is made up of a large number of emitters with low pulse repetition frequency (PRF) and some very high PRF emitters. Therefore the environment is very dense and signals from different emitters may be interleaved.

The EW receiver must deinterleave and group pulses from each individual radar. This is referred to as signal sorting. After sorting the signals into groups, the type of radar needs to be identified from the collected pulses. This is referred to as radar type identification.

In 1982, Davies and Hollands [1] presented automatic processing and decision-making techniques. Various approaches for deinterleaving are discussed, and relative merits of emitter parameters for each method are identified. In 1985, Whittall [2] discussed the problem of developing automatic sorting of radar signals (deinterleaving). The paper separates the signal sorting problem into two sub-problems of describing the radar environment in pulse-train sorter and monitoring changes in the perceived environment in a new-signal detector.

The first new algorithm in deinterleaving, presented by Mardia [3], was named cumulative difference histogram (CDIF). A cumulative time-of-arrival difference histogram gives an indication to probable pulse repetition intervals with a minimum number of computations. Validation and identification is given by searching for a sequence of these pulse intervals. The technique presented is less sensitive to interfering pulses and more robust to missed pulses than conventional published techniques.

In 1990, Anderson *et al.* [4] presented the first intelligent algorithm for radar pulses deinterleaving and identification using neural networks. The radar data were processed by a neural network designed to simplify the complex, as humans do, by breaking information into manageable blocks of data.

Four self-organising neural networks were compared by Granger *et al.* 1998 [5] for automatic deinterleaving of radar pulses streams in EW systems. The neural networks were the fuzzy adaptive resonance theory (FART), fuzzy min–max clustering (FMMC), integrated adaptive fuzzy clustering (IAFC) and self-organizing feature mapping (SOFM). Given the need for a clustering procedure that offers both accurate results and computational efficiency, these networks were examined from three perspectives – clustering quality, convergence time and computational complexity.

In this paper, a new scheme is presented for fast and accurate identification of several repetitive radar signals.

This scheme deinterleaves the signals using information other than TOA, such as direction-of-arrival (DOA), radio frequency (RF) and pulse width (PW). This technique groups signals that appear to belong to the same emitter together by applying cluster analysis. Then TOA analysis is applied on each group or emitter one by one.



doi:10.1049/iet-rsn:20070037

Paper first received 11th March and in revised form 11th April 2007

A.W. Ata'a is with the Computer Engineering Department, College of Engineering, University of Baghdad, Baghdad, Iraq, Jadiriyah, P.O. Box 47024

S.N. Abdullah is with the Electronic and Communication Engineering Department, College of Engineering, University of Baghdad, Baghdad, Iraq, Jadiriyah, P.O. Box 47024

E-mail: anaswasill@yahoo.com

## 2 EW receiving system

A simplified block diagram of an EW receiving system is shown in Fig. 1. This block diagram represents the functional blocks of the system. The first block is the feature extractor, which represents the antenna, RF receiver hardware and the parameter measurement and encoding circuitry. The output of the feature extractor is a pulse descriptor word (PDW) which contains the feature values of the intercepted signal [i.e. RF, amplitude (AMP), PW, TOA and DOA] [6].

Modern EW receiving systems must operate in an increasingly dense signal environment. Hence, a large quantity and jumble of signals are intercepted that must somehow be sorted in a timely and efficient manner so that subsequent actions, such as identifying the signal's origin, can be taken. The second block of Fig. 1 performs the sorting or deinterleaving function by clustering the incoming radar pulses into groups. In principle, each group or cluster should represent a single radar or emitter. However, the task of isolating a particular signal from a specific emitter can be difficult to accomplish, since the parameter boundaries between different signals may overlap, and since factors such as measurement error can cause the measured characteristics of the signal to become inexact or 'fuzzy' [6].

The final two blocks of Fig. 1 support the task of identifying and classifying the intercepted signals. The third block, the PRF-type Identifier, uses the sorted information from the deinterleaver to compute any patterns (e.g. the PRF pattern) that may be contained in each data cluster (emitter) by using the appropriate data item from the PDWs stored in a cluster. It also acts as a long-term memory for the clusters that were formed by the deinterleaver. The final and fourth block is the classifier, which ascribes each data cluster to a particular emitter, and which assigns (if appropriate) a priority to each emitter that is based on the presumed lethality of that emitter.

The identity of a particular signal is usually determined by correlating the observed characteristics of that signal with those that are stored in the Electronic Order of Battle (EOB), which is a list that contains the identity and signal characteristics of all known radars or those likely to be encountered. Since the evidence is often inexact or ambiguous, and since the EOB is finite in content, the inference process inherently contains some degree of uncertainty.

## 3 Deinterleaving

The deinterleaving algorithm tries to separate the received pulses according to their origins. Since each emitter has its own DOA and RF, clustering algorithms can be used to separate the emitters according to their features in the DOA–RF space.

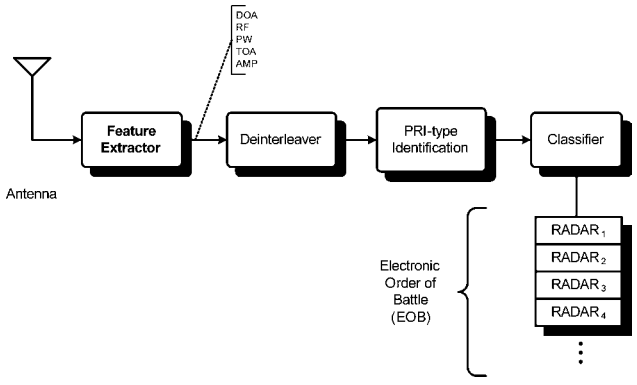


**Fig. 1** *EW system*

### 3.1 Cluster analysis

A number of techniques exist for separating patterns in two or more dimensions. These techniques are collectively called cluster analysis, and are described in [7, 8]. For a typical cluster analysis algorithm, the number of computational operations involved in carrying out the cluster analysis increases approximately as the square of the number of data patterns to be clustered [9].

A clustering method suitable for radar EW system should have the following properties. First, it should not require prior knowledge of the number or characteristics of categories to be found. Second, since the variable input arrival rate may reach $10^6$ patterns per second, it should be able to cluster non-stationary streams of input patterns sequentially, without requiring their long-term storage. Lastly, the sequence of operations needed for implementing the method using current technologies should lend itself well to high-speed hardware realisations. Given that, most of the popular well-established classical and fuzzy clustering algorithms require prior knowledge on either the number or the characteristics of cluster sought. Moreover, they need to do several iterations with the data set, or store the entire data set in memory [5]. None of these algorithms is suitable for the radar interception problem and therefore will not be discussed here.

A good solution is to use the self-organising neural networks (SONN). A SONN that is based on competitive learning was selected for this implementation; it is called 'Fuzzy Adaptive Resonance Theory (Fuzzy ART)'. This clustering algorithm can deal with modern environments.

*3.1.1 Modern deinterleaving:* The Fuzzy ART clustering has several advantages in solving modern clustering problems; the first one is that the Fuzzy ART clustering is independent of data presentation order for clearly separated non-overlapped clusters. This feature makes the algorithm insensitive to parameter measurement noise such as missing pulses or superior pulses. The other one is that the clusters shape in the Fuzzy ART clustering is not fixed (i.e. circular or rectangular); it can change adaptively with the cluster shape provided that the cluster size cannot exceed the maximum allowable limit (as will be seen in the next sections). This feature enables the algorithm to adapt to changing radar environments and track moving emitters.

Finally, setting the algorithm to fast learning mode ($\beta \rightarrow 1$ in the next sections) will converge in most cases from the first presentation of the data. This will make the clustering algorithm run in real-time. Fast adaptable and efficient clustering will be essential to relieve load on the TOA analysis in the next process.

For overlapped and/or very close clusters case, the Fuzzy ART algorithm will separate the large overlapped clusters into several clusters. However, the shape of these clusters depends mainly on cluster size and data presentation order. The clustering quality in this case will be reduced.

### 3.2 Fuzzy ART structure

The general structure of the Fuzzy ART neural network is shown in Fig. 2. It consists of two layers of neurons that are fully connected: a $2M$ neurons input or comparison layer ($F_1$), and an $N$ neurons output or competitive layer

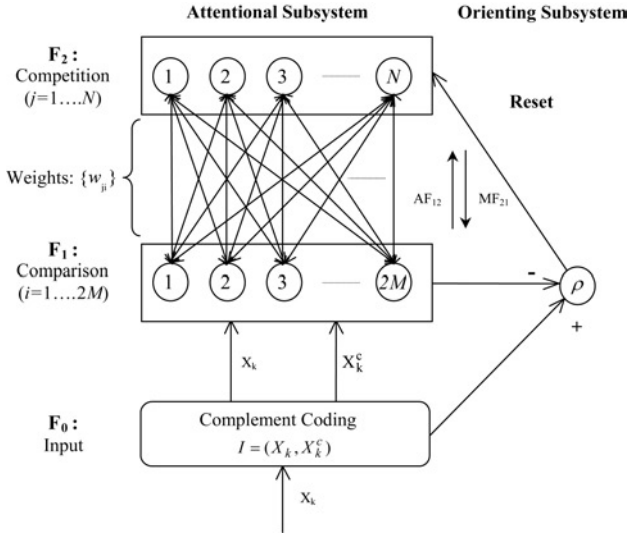


**Fig. 2** *Fuzzy ART structure*

($F_2$). where $M$ is the number of input dimensions and $N$ is the number of output categories.

The input layer ($F_0$) has no neurons; the function of the input layer is to normalise the input pattern and calculating its complement using complement coding. A weight value $w_{i,j}$ is associated with each connection between the ($F_1$) and ($F_2$) layers, where the indices $i$ and $j$ denote the neurons that belong to the layers ($F_1$) and ($F_2$), respectively.

Given an input vector to the ART network, $\boldsymbol{X}_i$, where $i = 1, \ldots, k$ and $k = 2M$, $M$ is the number of dimensions. The input layer ($F_0$) calculates the intermediate input **I**.

$$\boldsymbol{I} = (\boldsymbol{X}_K, \boldsymbol{X}_K^c) \tag{1}$$

The activation function AF for each of the PE's in the $F_2$ layer is defined as

$$\mathrm{AF}_j = \frac{|\boldsymbol{I} \Lambda \boldsymbol{W}_J|}{\alpha + |\boldsymbol{W}_j|}, \quad j = 1, \ldots, N \tag{2}$$

The size of a vector ($|\boldsymbol{X}|$) is determined by its $L_1$-norm (The $L_1$-norm is definded by $|\boldsymbol{X}|^{(r)} = r\sqrt{(\sum_{i=1}^{m} x_i^r)}$, $r = 1$.), the sum of its components. $\Lambda$ is the fuzzy conjunction, which is defined.by $z\ \Lambda y = \min\{x, y\}$. The choice parameter $\alpha$ provides a floating-point overflow, if $|\boldsymbol{W}_j| \to 0$. In some additional properties of fuzzy ART with variations on $\alpha$ are pointed out such as lowest possible vector size of prototypes. Simulations in this paper are performed with a value of $\alpha = 0.01$ The vector $\boldsymbol{W}_j = W_{1j}, \ldots, W_{dj}$, is the bottom-up connection weights between the $d$ neurons of $F_1$ layer and the $j$th neuron of $F_2$ layer. The function AF then provides a similarity measure between the input vector $\boldsymbol{X}$ and the class template (or cluster prototype) $\boldsymbol{W}_j$. The PE in $F_2$ with highest AF will be the winning neuron. The converse top-down match function that measures the similarity between the class template $\boldsymbol{W}_j$ and the input vector $\boldsymbol{X}$ is

$$\mathrm{MF}_j = \frac{|\boldsymbol{I} \Lambda \boldsymbol{W}_j|}{|\boldsymbol{I}|}, \quad j = 1, \ldots, N \tag{3}$$

The winning processing element $j$ that satisfies

$$\mathrm{MF}_j > \rho, \quad \rho \in [0, 1] \tag{4}$$

is then in the resonance domain. The parameter $\rho$ is a user-defined value indicating the vigilance of the network. The weight values of the processing elements that satisfy the $\rho$ test will then be updated towards the new pattern X using

$$\boldsymbol{W}_j^{(\mathrm{new})} = \beta \cdot (\boldsymbol{I} \Lambda \boldsymbol{W}_j^{(\mathrm{old})}) + (1 - \beta) \boldsymbol{W}_j^{(\mathrm{old})} \tag{5}$$

where $\beta$ is the learning rate. If the winning PE does not satisfy the $\rho$ test, an other PE with the next maximum AF will be considered as the winning neuron. If no PE satisfies the $\rho$ test, a new PE in $F_2$ will be generated with its initial weight values equal to $\boldsymbol{I}$ component values.

The set of weights $\boldsymbol{W}$ encodes information that defines the categories learned by the network. These can be modified dynamically during network operation. For each neuron $j$ of $F_2$, the vector of adaptive weights $\boldsymbol{W}_j = (w_{j1}, w_{j2}, \ldots, \boldsymbol{W}_{j2M})$ corresponds to the subset of weights $\boldsymbol{W}_j \subset \boldsymbol{W}$ connected to neuron $j$. This vector is called prototype vector, and it represents the set of characteristics defining the category $j$. Each prototype vector $\boldsymbol{W}_j$ is formed by the characteristics of the input patterns to which category $j$ has previously been assigned through winner-take-all competitions.

The learning rate $\beta \in [0, 1]$ defines how quickly prototypes converge to the common minimum of all input patterns assigned to the same cluster. With $\beta \to 1$, the network is working in a fast learning mode, stabilising the network after a few presentations of all training patterns (few epochs). In contrast, lower learning rates lead to a slow learning mode. ART-networks can simply be run in a pure classification mode by setting the learning rate of previously trained network to zero, which prevents all prototypes from being modified by new input patterns

A useful method to accelerate learning in ART networks is to set the learning rate $\beta = 1$ whenever a previously uncommitted cluster is adapted to the current input vector. The input $\boldsymbol{I}$ is identically copied as the first prototype of a new cluster if no other stored prototypes matches the input well enough. Committed prototypes might then be adapted more slowly ($\beta < 1$), to preserve them from being corrupted by noisy input patterns. This method is called fast-commit slow-recode.

Using complement coding, (3) is reduced to

$$\rho \leq \frac{|\boldsymbol{I} \Lambda \boldsymbol{w}_j|}{M} \tag{6}$$

Working in fast learning mode $\beta \to 1$, a prototype $\boldsymbol{W}_j$ in Fuzzy ART represents the common MIN-vector of all input $l$ patterns $\boldsymbol{I}_\mathrm{P}$, with p = 1, …, $l$, assigned to the same cluster $j$, in at least one presentation

$$\begin{aligned} \boldsymbol{W}_j &= \min(\boldsymbol{I}1, \boldsymbol{I}2, \ldots, \boldsymbol{I}l) \\ &= [\min(\mathrm{i}11, \ldots, \mathrm{i}11), \ldots, \min(\mathrm{i}1\mathrm{k}, \ldots, \mathrm{i}l\mathrm{k})] \end{aligned} \tag{7}$$

Using complement coding [10], input patterns $\boldsymbol{I}_\mathrm{P} = (\boldsymbol{A}_\mathrm{P}, \boldsymbol{A}_\mathrm{P}^\mathrm{C})$ lead to prototypes representing the common MIN and MAX vectors of all uncoded patterns $\boldsymbol{A}_\mathrm{P}$

$$\begin{aligned} \boldsymbol{W}_j &= (\min\{\boldsymbol{A}_1, \ldots, \mathrm{A}_l\}, \min\{\boldsymbol{A}_1^\mathrm{c}, \ldots, \mathrm{A}_l^\mathrm{c}\}) \\ &= (\min\{\boldsymbol{A}_1, \ldots, \mathrm{A}_l\}, \max\{\mathrm{A}_1, \ldots, \mathrm{A}_l\}^\mathrm{c}) \end{aligned} \tag{8}$$

With lower learning rates $\beta$, networks prototypes converge more slowly to these MIN and MAX bounds.

In other words, $\boldsymbol{W}_i$ for $i \geq M$ is still the cluster minimum (assuming fast learning) for the $i$th variable, but $1 - w_{(i+M)}$ is the cluster maximum. Thus each cluster seed represents both cluster minimum and cluster maximum and can be thought of as a box within the hyper-cubic data space. Fig. 3 shows an example of two-dimension cluster seed.

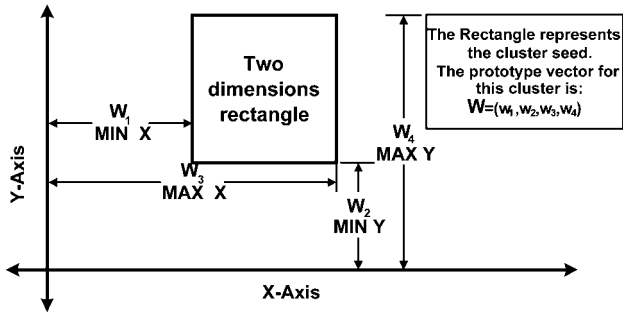


**Fig. 3** *Cluster seed example*

Complement coding does indeed prevent the number of categories from growing without bound, but it does nothing to ensure that the number of categories is appropriate in any practically useful sense.

With complement coding, the vigilance parameter places an upper bound on the size of each category box. The size of a box is defined as the sum of its dimensions

$$\text{size} = \sum_{i=1}^{M} [1 - w_{(i+M)} - w_i] \leq (1 - \rho) \tag{9}$$

During training, when a new case is assigned to a category, the box grows just enough to contain the new case. As the box grows, the set of patterns that resonate with the category shrinks. When the box reaches the size given by the right-hand side of the above formula only patterns within the box resonate with the category, hence the box cannot grow any larger and the category boxes can, and frequently do, overlap [11].

## 4 PRF-type identification

The next block of the system is the PRF-type identifier. Since the deinterleaver separates the incoming pulses into groups or clusters, the PRF-type will be extracted for each group. The only available pulse feature that is depending on the PRF pattern is the TOA. By analysing the TOAs of all pulses in a cluster or group; the PRF pattern may be estimated.

### 4.1 PRF categories

The variety of PRF schemes is seemingly infinite [12]. There are, however, a number of schemes used sufficiently often these categories have been given names. Generally, the radar emitter falls into one of the following three categories.

*4.1.1 Constant PRF:* Some times called regular PRF. The radar has nearly constant PRF if the peak PRF variations are less than about 1% of the mean PRF.

*4.1.2 Jittered PRF:* Jittered PRF is the random PRF variations of large amounts. Intentional jitter is used as electronic-counter-countermeasure (ECCM) against some types of jamming. The amount and type of jitter can also aid in identifying the type of radar transmission being received.

*4.1.3 PRF stagger:* A staggered pulse train is fundamentally a basic PRF with this same PRF impressed upon itself one or more times. Each level of impression (stagger) utilises a different start time or reference which will preclude the generation of concurrent pulses or pulses shadowing one another. The number of levels (or positions) is the number of times the basic PRF/IPP (inter-pulse period) is integrated in the pulse train. ('EW tutorial', https://www.myaoc.org/EWEB/dynamicpage.aspx?webcode = GoldCrows_EWTutorial2 #Sec2_2_3.). Generally, stagger is used to eliminate blind speeds in MTI radar systems.

Fig. 4 illustrates the time relationship involved in the generation of a three-level stagger. Each level has the same characteristic PRF and pulse width, but the Time to First Event (TFE) for each level is different. This has the effect of slewing the masked pulse groups in relation to one another resulting in the desired stagger pattern.

### 4.2 PRF-type identification algorithm

Each cluster or group may contain one or more emitters. If a group contains only one emitter, then the emitter is one of the following types: simple PRI, staggered PRI or jittered PRI. In case of multiple emitters, as the first emitter is detected its pulses are extracted. This extraction will make the process of detection of the remaining emitters easier.

Fig. 5 shows the PRF-type identification algorithm. The first step of processing is done by the TOA folding Histogram algorithm. If only one emitter is presented in the current group, this algorithm must succeed from the first check. This algorithm has an advantage against large amount of missing pulses. It will be passed only one time (checking for only one PRI). If the algorithm succeeds to identify this PRI, this PRI will be recorded and another

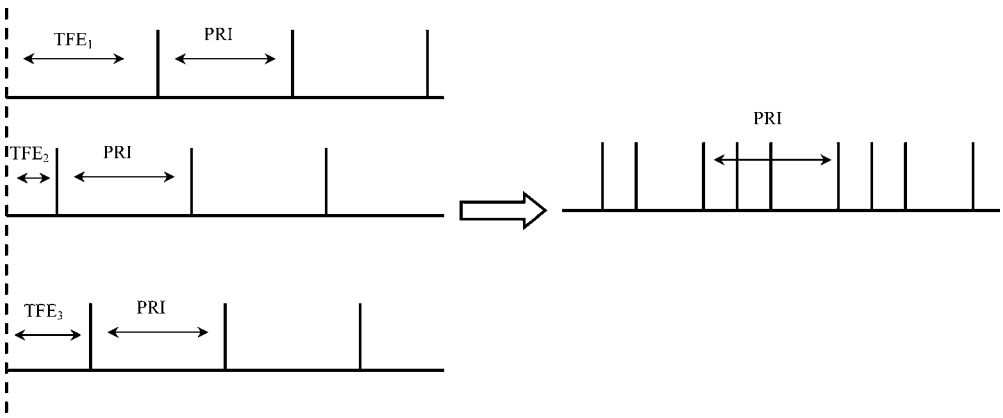


**Fig. 4** *Generation of staggered PRF*

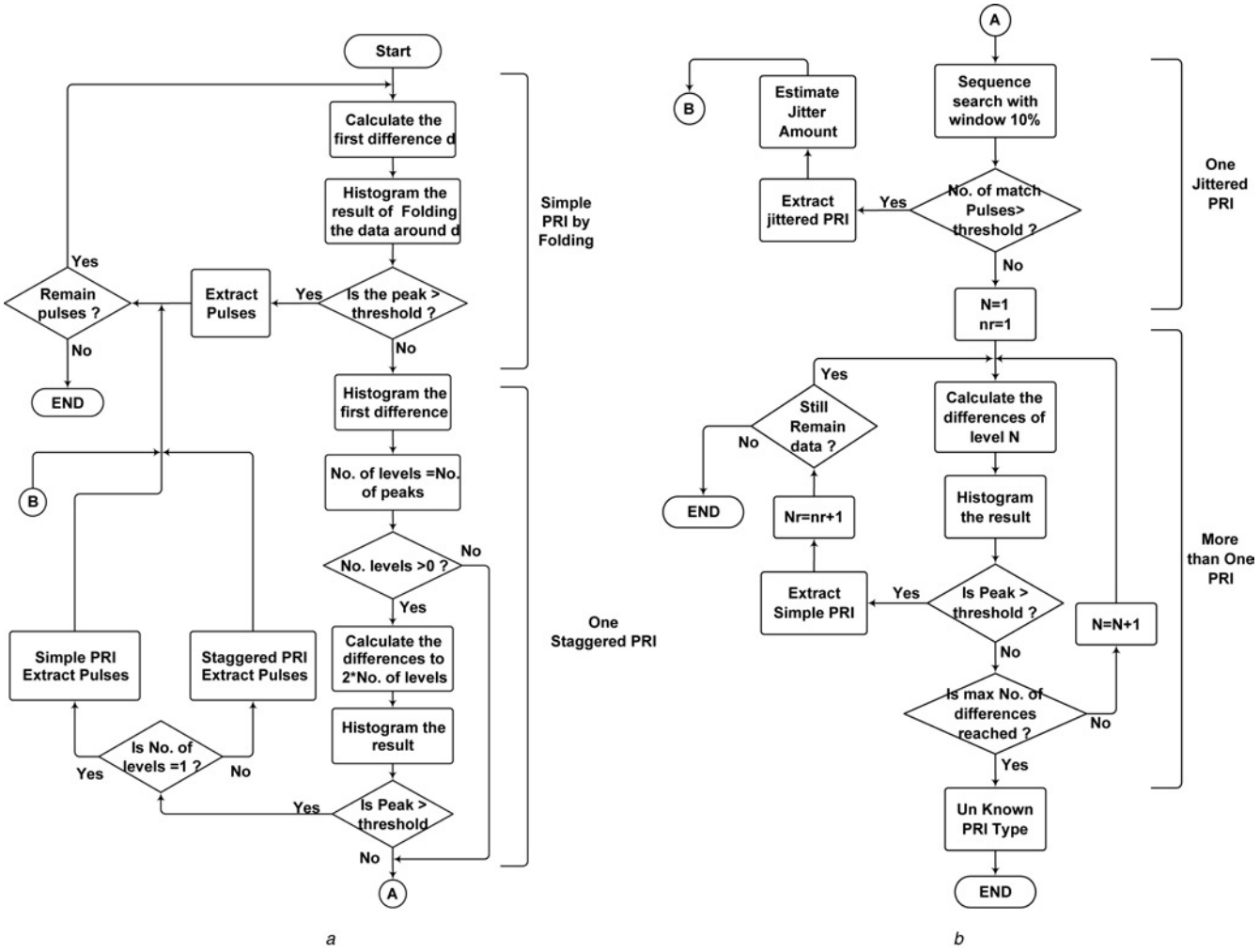


**Fig. 5** *PRF-type identification flowchart*

group is examined. The folding can extract only one simple PRI emitter, if more than one emitter presents in the cluster or the emitter is not simple PRI, the algorithm will fail and the data is passed to the next process. The algorithm is said to be successful when the number of pulses that lie within one bin is larger than a threshold. The threshold calculated by dividing the time of observation by the checked PRI value.

The next process is to check for stagger PRI. The first-level difference was used in this check. The stagger PRI will result in equal peaks, the number of these peaks equals to the number of stagger levels. From this observation, the number of levels is estimated. When more than one peak is detected, the PRI is calculated by applying the Differences Histogramming with the number of difference levels larger than the number of detected peaks. If only one peak is present then the emitter is not a stagger PRI, it is simple PRI but with noise. If no peaks are detected at all, then the emitter is jittered PRI.

Jittered PRI can be verified by using the Sequence Search Algorithm but with relative wide allowance window. If the number of matched pulses now exceeds the threshold, the emitter is jittered PRI and the jitter amount will be estimated. If the number of matched pulses does not exceed the threshold then either the cluster contains more than one emitter or the data is only noise.

More than one PRI is processed by the CDIF Histogram until an emitter is detected. Once an emitter is detected, its pulses are extracted and the remaining pulses pass the TOA analysis again. This process will stop if there are no remaining pluses or the data is corrupted.

## 5 Implemented system

The over all proposed system is shown in Fig. 6. The first block of the system is the clustering. In this block, the received radar signals are clustered using some of its parameters. These parameters are DOA and RF. The clustering is done through the FART (discussed in Section 3). This algorithm will separate the radars data into number of clusters or groups. Then by analysing these groups one by one, the characteristics of each group will be extracted. The group characteristic in this discussion is the PRF-type. The PRF-type detection and extraction algorithm is discussed in Section 4.

The PRF-type identification is not the final goal of the system. Each identified emitter should be ascribed to a known radar type. The classifier stage is beyond the scope of this paper.

## 6 Results

### *6.1 Fuzzy ART clustering*

The fuzzy ART algorithm had been implemented in matlab, and as C-mex file for speed up purposes. In the following sections the convergence time and clustering quality results will be presented.

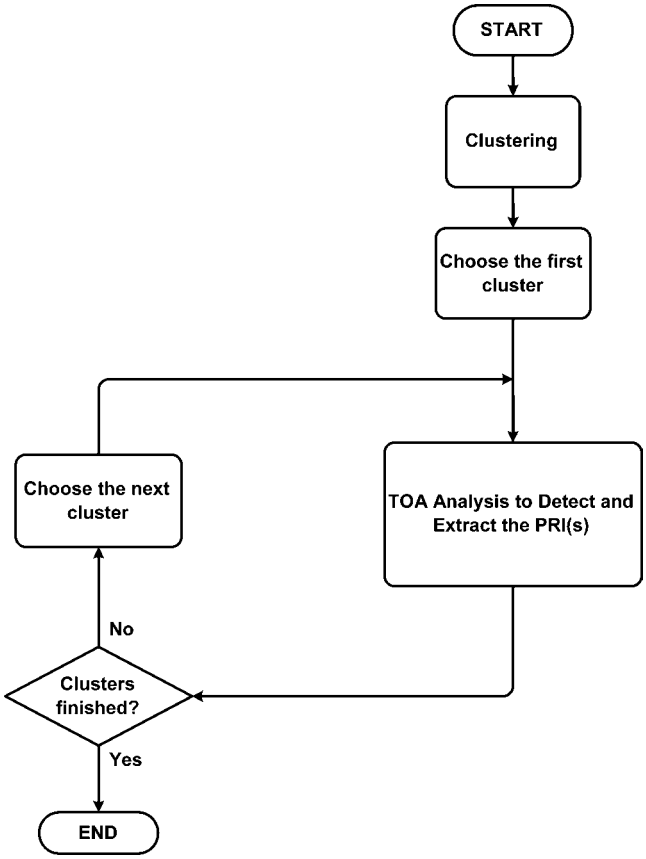


**Fig. 6** *Complete system*

*6.1.1 Convergence time:* During computer simulation, each complete presentation of an input data set to the SONN is called Epoch [5]. As each pattern is presented, the weights will converge towards an optimum value. Once convergence is reached, weights values remain constant during subsequent presentation of entire data set in any order. This change depends on the learning rate. The learning rate, $\beta$, can be within the range $0 \leq \beta \leq 1$. Fig. 7 shows the variation of the mean square (MS) weights error against input pattern number for different values of $\beta$. The MS weights error represents the summation of the square errors between all weights of output neurons and the corresponding target weights for all clusters.

$$\text{MS weights error} = \sum_{i_1=0}^{J} \sum_{i_2=0}^{2M} (W_{i_1 i_2} - T_{i_1 i_2})^2$$

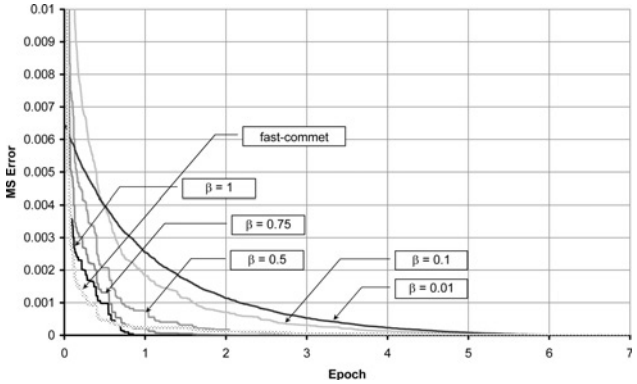


**Fig. 7** *Mean square error against epochs*

where $J$ is the number of clusters (output categories), $M$ is the number of input dimensions, $\boldsymbol{W}_i$ the weight vector, $T_i$ the target weight vector.

The tested data sets consist of five clusters, and $\rho$ value was set to 0.9.

In all of these tests the output of the network was five clusters with structure exactly the same as the true structure of the data patterns. Note that when $\beta = 1$, the network convergence remains stable in less than one epoch. However setting $\beta = 1$ reduces the clustering quality. On the other hand, smaller values of $\beta$ will give better but slower error curve. The data may be trained in non-real-time by passing the data to the algorithm more than one time (more than one epoch) with $\beta$ less than one; this will certainly improve the output clusters structure at the expense of increased convergence time.

*6.1.2 Clustering quality:* The clustering of the algorithm will be compared with the true cluster structure to give the clustering quality of the algorithm. Usually, the clustering algorithm is compared with a standard clustering algorithm such as the ISODATA algorithm. But since this package was not available, the algorithm is compared with the true clusters structure. This will expect to reduce the clustering quality slightly (similar to comparing with ideal clustering algorithm).

A partition of $n$ patterns into $K$ groups defines a clustering [5]. This can be represented as a set $A = \{a_1, a_2, \ldots, a_n\}$, where $a_h \in \{1, 2, \ldots, K\}$ is the category label assigned to pattern $h$. The degree of match between two clustering algorithms, say A and B, may be compared by constructing contingency table, as shown in Table 1. The four variables within the contingency table have been used to derive measure of similarity between two clustering algorithms A and B [7]. These measures are known in pattern recognition literature as external criterion indices [13].

An external criterion measure is used to evaluate recovery of true cluster structure. An external criterion index uses information obtained from outside the clustering process. In the present situation, the external information is the knowledge of the true cluster structure [13].

The four cells in the table indicate whether each pair of points was correctly classified by the algorithm or not. For example, cell '$a$' indicates the frequency count for the number of pairs which were correctly clustered by both the algorithm and the external criterion. On the other hand, cell '$c$' indicates the number of occurrence where the algorithm placed a pair of points in different clusters when the points actually came from the same cluster. Cell '$b$' is exactly the reverse situation, while cell '$d$' indicates the number of occurrence where the algorithm placed a pair of points in different clusters when the are actually came from different clusters. Thus, cells '$a$' and '$d$' indicate the frequency of correct pairwise classifications, and cells '$b$' and '$c$' represent the number of improperly clustered pairs. The sum of all elements $m$ is

$$m = a + b + c + d = \binom{2}{n} = \frac{n(n-1)}{2} \qquad (10)$$

**Table 1: Contingency table**

| | | Clustering A | |
|---|---|---|---|
| | | Same | Different |
| Clustering B | Same | *a* | *b* |
| | Different | *c* | *d* |

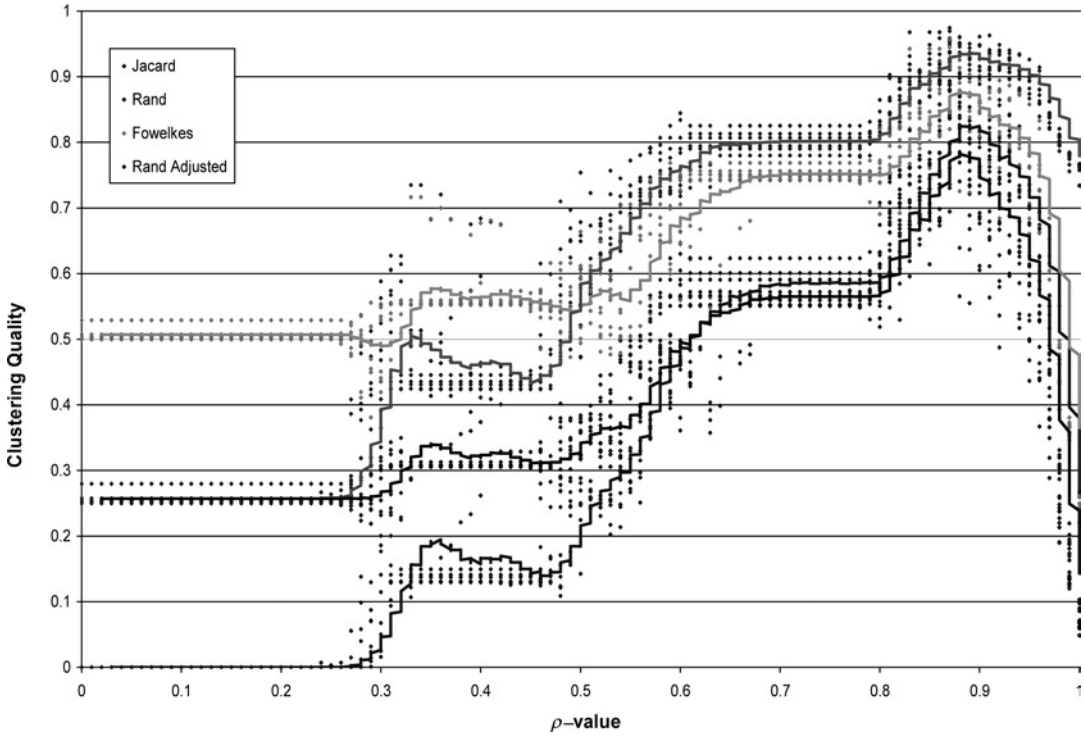


**Fig. 8** *ρ-test with random data*

It is the total number of combinations of two out of $n$ patterns. Given these definitions, the statistic used in calculating the clustering quality is called 'Rand Adjusted' and it is defined [7, 13] as follows

$$S_{\mathrm{RA}} = \frac{2(ad - bc)}{2ad + (a+d)(b+c) + b^2 + c^2} \tag{11}$$

As mentioned in the previous chapter, the algorithm needs to specify three constants $\alpha$, $\beta$, $\rho$ The constant $\alpha$ was set to 0.001 in all the simulations since it has no effect on the performance. The second parameter $\beta$ was set to 1. The third parameter determines the granulation of the clusters; therefore accurate value is important to reconstruct the clusters structure correctly. Values between 0 and 1 were tested in two different cases. In the first case, the data were presented in random order (Fig. 8), whereas in the second case the data were presented in bursts (Fig. 9). These two cases are the most common data order in practical systems (Granger et al 1997). The values of $\rho$ starts from 0 to 1 with step size of 0.01, ten tests were used for each step (total no. of 1000 tests). Each test consists of five clusters, with pattern deviation in each cluster chosen randomly between 0.005 and 0.015. The number of patterns in each cluster was chosen randomly between 10 and 60. It is clear from these figures that the value of 0.88 for $\rho$ is the value of the highest clustering quality for the cluster shapes used.

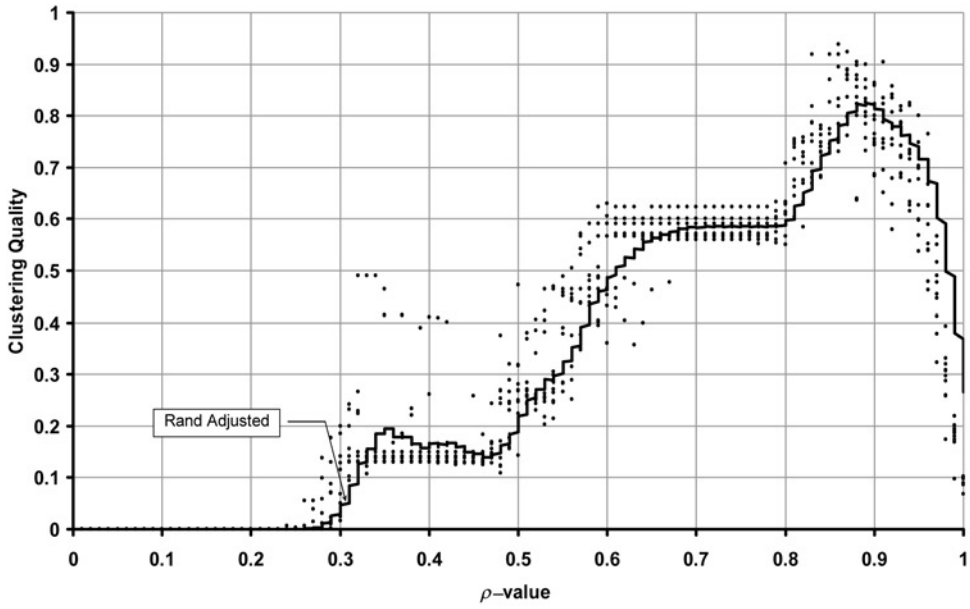


**Fig. 9** *ρ-test with burst data*

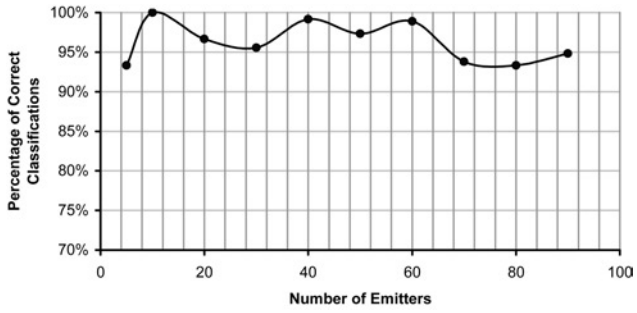


**Fig. 10** *Simple PRI test classifications*

In another 1000 different tests for $\rho = 0.88$, the average clustering quality was 0.825703 and the average deviation was 6.470%.

### 6.2 Complete system tests

The complete proposed system described in Section 4.5 has been implemented and tested. In the following sections, some of the tests will be presented.

*6.2.1 Simple PRI test:* The complete system is tested for simple PRIs. Different number of emitters from 5 to 100 had been tested. Fig. 10 shows the results of this test. The $y$-axis is the percentage of correct classifications and the $x$-axis is the number of emitters. It can be seen that the complete system works properly in the case of simple PRI emitters for very large number of emitters (up to 100).

*6.2.2 Staggered PRI test:* Fig. 11 shows the percentage of correct classifications against the number of emitters for up to 30 emitters. The percentage of correct classifications now decreasing rapidly when the number of emitters increased.

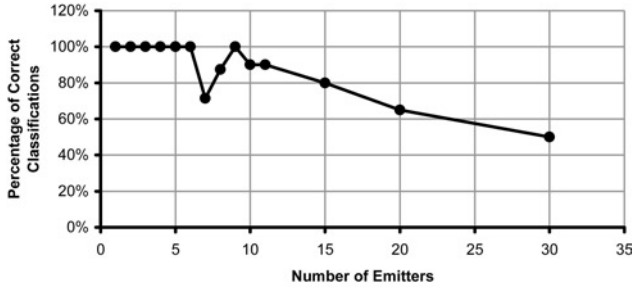


**Fig. 11** *Staggered PRI classifications*

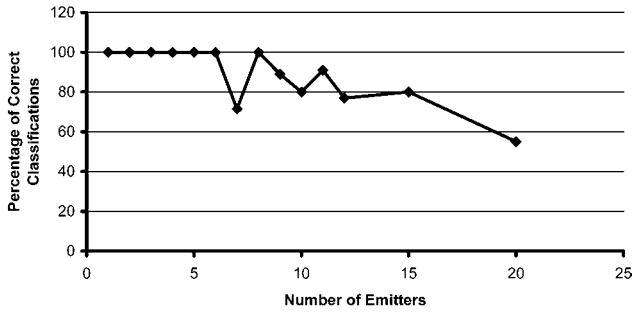


**Fig. 12** *Jittered PRI classifications*

*6.2.3 Jittered PRI test:* The jittered PRI test shown in Fig. 12. The percentage of correct classifications decreases faster than that of staggered PRI.

## 7 Conclusion

The radar signals have been identified using an intelligent system based on ART of neural network. It has the ability to handle large number of emitters at the same time as large as 90 simple emitter and few tens complex emitters.

The Fuzzy ART used in clustering has a high clustering quality about 90%. The algorithm can run in real-time with relatively small memory to store the weights only. The clustering quality depends on the vigilance parameter $\rho$. Also clustering quality depends on the learning rate $\beta$. Small values of $\beta$ give better clustering quality but requires more epochs (more time). The Convergence time of this algorithm is very fast. Fast learning rate ($\beta = 1$) requires only one epoch to converge in most cases.

Several algorithms can be used to detect the presence of a certain PRI in a received data based on TOA of the received pulses. Some of these algorithms are TOA Folding Histogram, DTOA Histogram and Periodogram. A new algorithm made of combination of these algorithms can identify simple, stagger and jittered PRIs.

## 8 Acknowledgment

This work is summarised from the work on a thesis submitted to the College of Engineering, University of Baghdad, in partial fulfilment of the requirements for the Degree of Master of Science in Electronic and Communication Engineering in 2002 [14].

## 9 References

1 Davies, C.L., and Hollands, P.: 'Automatic processing for ESM', *IEE Proc., Part F*, 1982, **129**, (3), pp. 164–171
2 Whittall, N.J.: 'Signal sorting in ESM systems', *IEE Proc., Part F*, 1985, **132**, (4), pp. 226–228
3 Mardia, H.K.: 'New technique for the deinterleaving of repetitive sequences', *IEE Proc., Part F*, 1989, **136**, (4), p. 149
4 Anderson, J.A., Gately, M.T., Penz, P.A., and Collins, D.R.: 'Radar signal categorization using a neural network', *Proc., IEEE*, 1990, **78**, (10), pp. 1646–1657
5 Granger, E., Savaria, Y., Lavoie, P., and Cantin, M.: 'A comparison of self-organizing neural networks for fast clustering of radar pulses', *Signal Process.*, 1998, **64**, (3), pp. 249–269
6 Vaccaro, D.D.: 'Electronic warfare receiving systems' (Artech House, 1993)
7 Anderberg, M.R.: 'Cluster Analysis for Applications' (Academic Press, 1973)
8 Hartigan, J.A.: 'Clustering algorithms' (Wiley, 1975)
9 Wilkinson, D.R., and Watson, A.W.: 'Use of metric techniques in ESM data processing', *IEE Proc., Part F*, 1985, **132**, (4), pp. 229–232
10 Frank, T., Kraiss, K., and Kuhlen, T.: 'Comparative analysis of Fuzzy ART and ART-2A network Clustering Performance', *IEEE Trans. Neural Netw.*, 1998, **9**, (3), pp. 544–559
11 Sarle, W.S.: 'Why statisticians should not FART', Cary, NC, USA, 1995, ftp://ftp.sas.com/pub/neural/fart.txt
12 Wiley, R.G.: 'Electronic intelligence: the analysis of radar signals' (Artech House, 1985)
13 Milligan, G.W., Soon, S.C., and Sokol, L.M.: 'The effect of cluster size, dimensionality, and the number of clusters on recovery of true cluster structure', *IEEE Trans. Pattern Anal. Mach. Intell*, 1983, **5**, pp. 40–47
14 Ata'a, A.W.: 'Radar PRF identification and clustering algorithms', MSc thesis, Baghdad University, 2002