A NEW ALGORITHM FOR THE DEINTERLEAVING OF RADAR PULSES

M. Kahrizi, E. Kabir and H. R. Bakhshi

Department. of Electrical Engineering
Tarbiet Modarres University
Tehran, Iran

Abstract This paper presents a new algorithm for the deinterleaving of radar signals, based on the direction of arrival (DOA), carrier frequency (RF), and time of arrival (TOA). The algorithm is applied to classic (constant), jitter, staggered, and dwell switch pulse repetition interval (PRI) signals. This algorithm consists of two stages. In the first stage, a Kohonen neural network clusters the received pulses on the basis of frequency and DOA. In this stage radars having the same frequency and DOA, are identified as one class. In the second stage, the number of existing emitters and their PRIs is determined by using TOA information. The algorithm for the deinterleaving uses the information obtained from the previous analysis to reduce the required computation time. The simulation results show that the algorithm is successful in high pulse density environments and for the complex signal types.

Key Words Radar, Kohonen Network, Clustering, PRI, Deinterleaving

چکیده امروزه به علت وجود تعداد قابل ملاحظه ای رادار در محیط های مورد مطالعه، لزوم یک الگوریتم کارآمد برای بازشناسی منابع راداری با استفاده از سیگنال های دریافتی توسط یک گیرنده به خوبی مشهود است. الگوریتم پیشنهادی دراین مقاله برای سیگنال های پالسی طراحی شده است و براساس سه پارامتر فرکانس، زاویه و زمان ورود سیگنال ها عمل می کند. در مرحلهٔ اول، با استفاده از شبکه عصبی کهنن و براساس فرکانس و زاویه ورود پالس ها که مقادیر نسبتاً پایدارتری هستند، کلاس هایی به عنوان خروجی تشکیل می شود. دراین مرحله اگر چند منبع دریک فرکانس و زاویه ورود وجود داشته باشند، به عنوان یک کلاس درنظر گرفته می شوند. در مرحلهٔ دوم، براساس زمان ورود پالس ها به گیرنده، تعداد منابع در هر کلاس و نوع مدولاسیون آنها مشخص می شود.

INTRODUCTION

Deinterleaving of radar pulses is a process of recognizing active radar emitters. Deinterleaving algorithms are usually based on the analysis of various parameters of the received radar pulses, such as palse amplitude, pulse width, carrier frequency, direction of arrival (DOA), and time of arrival (TOA). Pulse amplitude and width are generally not very useful because they often vary from pulse to pulse due to intentional variation by the emitter or due to multipath effects [1]. DOA and frequency are generally regraded as the best parameters since they can not be varied by the emitter from pulse to pulse and could be considered to be stable.

Deinterleaving algorithms based on TOA generally use the histogram of TOA pulses [2,3]. In these algorithms, if there are many emitters or the nature of the received signals is too complex, the identification is usually hard and could be impossible. Another restriction in these methods is that if some of the pulses have the same TOA, they are considered as a single pulse.

The algorithm for the deinterleaving presented in this paper uses the information about DOA, carrier frequency, and TOA of the received pulses. The major contribution of this work is the effectiveness of the TOA processing in a complex environment. In

TOA analysis, first the different types of possible pulse repetition intervals (PRIs) have been extracted with the associated numbers. Then the logical relations between them are needed to meet some special conditions. The different combinations of the remained pulses are checked to achieve the final results. However, simple cases only require the primary steps, hence reducing the time considerably.

I. ALGORITHM DESCRIPTION

In this section, the major steps of the algorithm are outlined. Clustering by DOA and RF are discussed first. Then, the steps which are considered for the deinterleaving according to TOA are presented.

Neural Network Clustering Based on DOA and RF

To determine the characteristics and the number of clusters which can be found from the received pulses, we have used the Kohonen neural network [4,5]. Since no information was available about the emitters, an unsupervised neural network would be a logical choice.

The network has two input neurons, one for the frequency and the other for the DOA. In order to estimate the number of output neurons, a simple threshold based on clustering algorithm is used [6]. If this algorithm yields M clusters, the output neurons of the Kohonen network is arranged in an (M+1) by (M+1) matrix, which provdies enough space for the clusters to grow.

The network is trained for an input frame of n pulses as follows: the input vectors are normalized. Let w_{ij}(t) be the weight for the connection between ith input neuron and jth output neuron, where $1 \leq i \leq 2, 1 \leq j \leq (M+1)(M+1)$ and t is the iteration number. $w_{ij}(0)$'s are randomly selected between 0 and 1, the initial neighborhood radius $N(0)$ is considered to be M, and the initial learning rate $\eta(0)$ is 0.5.

In each iteration, the Euclidean distance between the input vector $(x_1(t), x_2(t))$ and the weight for the output neurons is calculated as

$$d_j^2 = (x_1(t) - w_{1j}(t))^2 + (x_2(t) - w_{2j}(t))^2 \quad (1)$$

The nearest neuron is considered as the winner (j^*). Then, the wieghts are adjusted as follows:

$$w_{ij}(t+1) = \begin{cases} w_{ij}(t) + \eta(t)[x_i(t) - w_{ij}(t)] & \text{if } |j^* - j| \leq N(t) \\ w_{ij}(t) & \text{if otherwise} \end{cases} \quad (2)$$

During the learning phase, $N(t)$ and $\eta(t)$ are adjusted as shown in Figures 1 and 2. When the weights converge within a predefined threshold, the training is ended. The weighted mean of the nodes in each cluster is taken as a prototype for that cluster.

As an example, a scenario consisting of five emitters with the nominal frequency and DOA listed in Table 1 is used for the training of the network. Taking account of noise and measurement errors, the value of frequency and DOA which are applied to the network are assumed to have Gaussian distribution

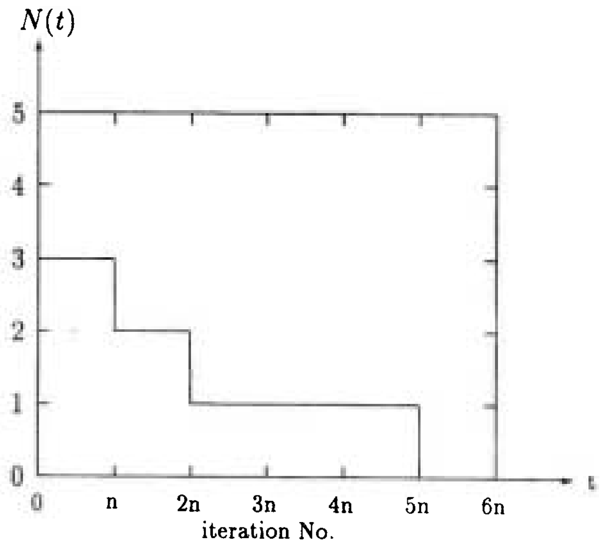

Figure 1. Neighborhood radius during the learning period

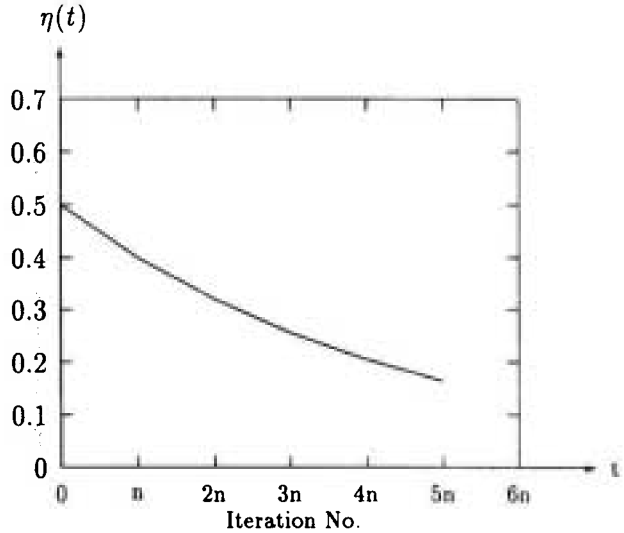

Figure 2. Learning rate during the learning period

TABLE 1: The Nominal Frequency and DOA, σ_f= .25 GHz and σ_{DOA}= 3°

Frequency (GHz)	DOA (degree)
8.25	15.3
11.7	40.6
10.5	30.5
11.2	20.2
12.1	25.7

[7] around those nominal values in Table 1. σ_f and σ_{DOA} in Table 1 are the variances of the frequency and DOA, respectively. These variances can be evaluated approximately by the same analysis used in [7].

The value of the weights at the end of the training phase are shown in Table 2. The first number in each box is frequency and the second number is DOA. c_i shows the weight which belongs to the *i*th class. The center of each cluster is calculated and shown in Table 3.

In the recall phase, the Euclidean distance between the input pattern and the prototypes for the existing clusters are computed. The input pulse is assigned to the cluster with the minimum distance if this distance

TABLE 2: Output Neurons at the End of a Training Phase for the Example of Table 1.

8.19 14.94 c_1	8.19 14.94 c_1	10.35 18.12	11.23 19.33 c_2	11.60 21.74	12.13 25.10 c_3	12.14 25.12 c_3
8.30 15.33 c_1	8.30 15.33 c_1	9.74 17.68	11.14 19.85 c_2	11.53 23.52	12.06 25.81 c_3	12.08 25.81 c_3
8.35 15.54 c_1	8.35 15.54 c_1	9.74 18.29	11.14 20.67 c_2	11.53 23.18	12.05 26.16 c_3	12.08 26.30 c_3
8.90 19.04	8.90 19.04	9.81 18.75	11.13 20.79 c_2	11.52 23.24	11.94 31.21	11.96 31.45
10.49 29.19 c_4	10.52 29.65 c_4	10.82 25.97	10.86 25.51	11.03 25.55	11.73 39.75 c_5	11.76 40.72 c_5
10.38 30.03 c_4	10.43 30.55 c_4	10.56 30.64 c_4	10.62 30.66 c_4	11.26 36.97	11.67 41.38 c_5	11.67 41.38
10.33 30.40 c_4	10.41 30.74 c_4	10.57 30.80 c_4	10.62 30.66 c_4	11.25 36.93	11.63 41.59 c_5	11.63 41.59 c_5

TABLE 3: The Center of the Output Clusters for the Example of Table 1.

Class No.	Frequency (GHz)	DOA (degree)
1	8.25	15.20
2	11.18	20.08
3	12.09	25.77
4	10.49	30.35
5	11.71	40.58

is less than a threshold. Else the input pulse is identified as a new emitter and the training starts with the new received frame.

It should be noted here that if the neighboring emitters are not completely disjoint in DOA or frequency, they may be assigned to the same cluster. However, in the TOA analysis, they will be identified as disjoint emitters.

Deinterleaving Based on TOA

The next task is to compute the pulse repetition pattern of an emitter by using the times of arrival for the pulses in a given cluster. This information will be used for emitter classification. The algorithm in this part can extract any of constant, jitter, staggered, or dwell switch PRI patterns.

First, we describe the scheme for extracting the possible PRI pattern from the pulses in every given cluster. Note that there exist many common pulses in the PRI patterns. Second, we combine all the obtained information about PRI patterns to identify possible emitters. Some of the subsets of information could cluster the emitters with a given degree of confidence. Addition of previous information can improve performance to some extent. Assuming that the pulses contained in a given cluster form a train of frames with N pulses, the basic steps are described below:

1) *Pattern Extraction for the Constant PRI:* It is known that for an emitter with constant PRI of period T and the starting time of T_{start}, TOA of the pulses is in the form of $(T_{start} + T, T_{start} + 2T, T_{start} + 3T, \ldots)$. Therefore, if the frame contains an emitter only with a constant PRI, by calculating TOA difference between any two adjacent pulses, called the first difference, the PRI can be identified. Also by finding the difference between TOA of the first pulse and the computed PRI, the starting time, T_{start}, can be obtained. However in real environment there can be many patterns in each frame. Also, the two adjacent pulses could be from different patterns which are interleaved based on their TOAs.

Now, let $t[i]$ and $t[j]$ be the TOAs of the first and the second pulse respectively, of an emitter with a constant PRI. Then, the algorithm looks for a group of pulses that form a periodical pulse train, with the periods T and the starting time of T_{start} by the following relations

$$T = t[j] - t[i] \tag{3}$$

$$T_{start} = t[i] - T \tag{4}$$

The number of pulses of this so called emitter, n, can be computed by

$$n = \frac{t[N] - T_{start}}{T} \tag{5}$$

To determine the constant PRI pulse train with $t[i]$ and $t[j]$ as TOAs of the first and the second pulses in this train, we have used the algorithm shown in Figure 3. The process is repeated until the extraction of all possible constant pulse trains from the original frame.

To improve the reality of the simulated data and the performance of the algorithm, It is assumed that x percent of the pulses in the ideal pulse train could be lost. Then a threshold P_{tr} of the total pulses in the frame has been checked in the algorithm. P_{tr} can be chosen as

$$P_{tr} \leq \alpha \tag{6}$$

where

$$\alpha = \frac{100 - x}{100} \tag{7}$$

To improve the accuracy, we can choose $P_{tr} < \alpha$, but this increases the processing time.

2) *Pattern Extraction for the Jitter PRI:* Jitter analysis is performed like the constant PRI. Therefore, the flowchart of Figure 3 has been utilized in this section with a small modification. The only difference is to replace $kT + T_{start}$ with

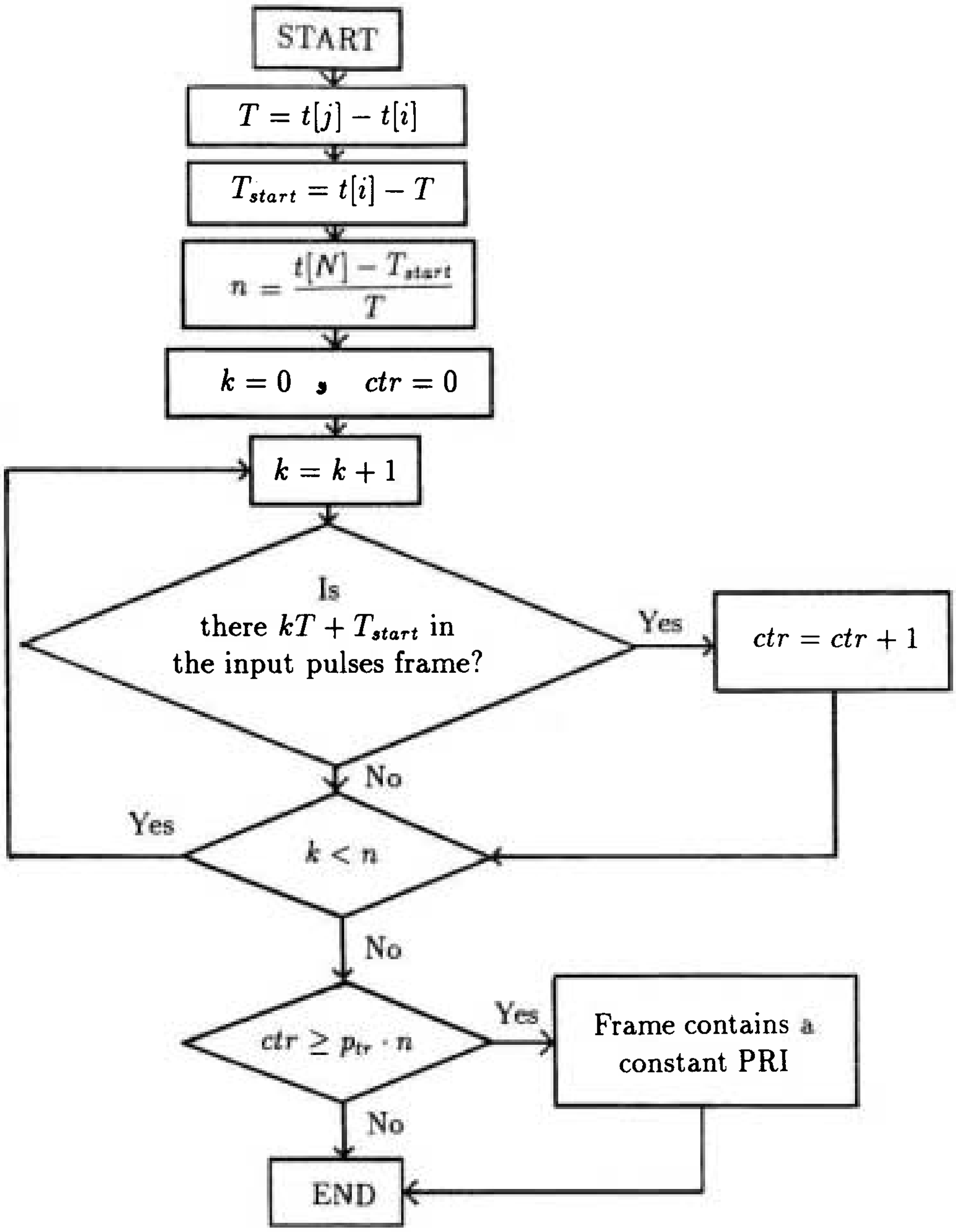

Figure 3. Algorithm for extracting constant PRI pattern

the nominal value of the deflection given for the jitter.

3) *Pattern Extraction for the Staggered PRI:* In the staggered PRI of the level two, the value of two PRI's $T[1]$ and $T[2]$ are specified. In this pattern the TOAs of the pulses can be considered as ($T_{start} + T[1]$, $T_{start} + T[1] + T[2]$, $T_{start} + 2T[1] + T[2]$, ...). If $t[i]$, $t[j]$, and $t[k]$ are the TOAs of the first, second, and third pulses, respectively, coming from an emitter with the aforementioned staggered pattern, then we can write

$$T[1] = t[k] - t[j] \tag{8}$$

$$T[2] = t[j] - t[i] \tag{9}$$

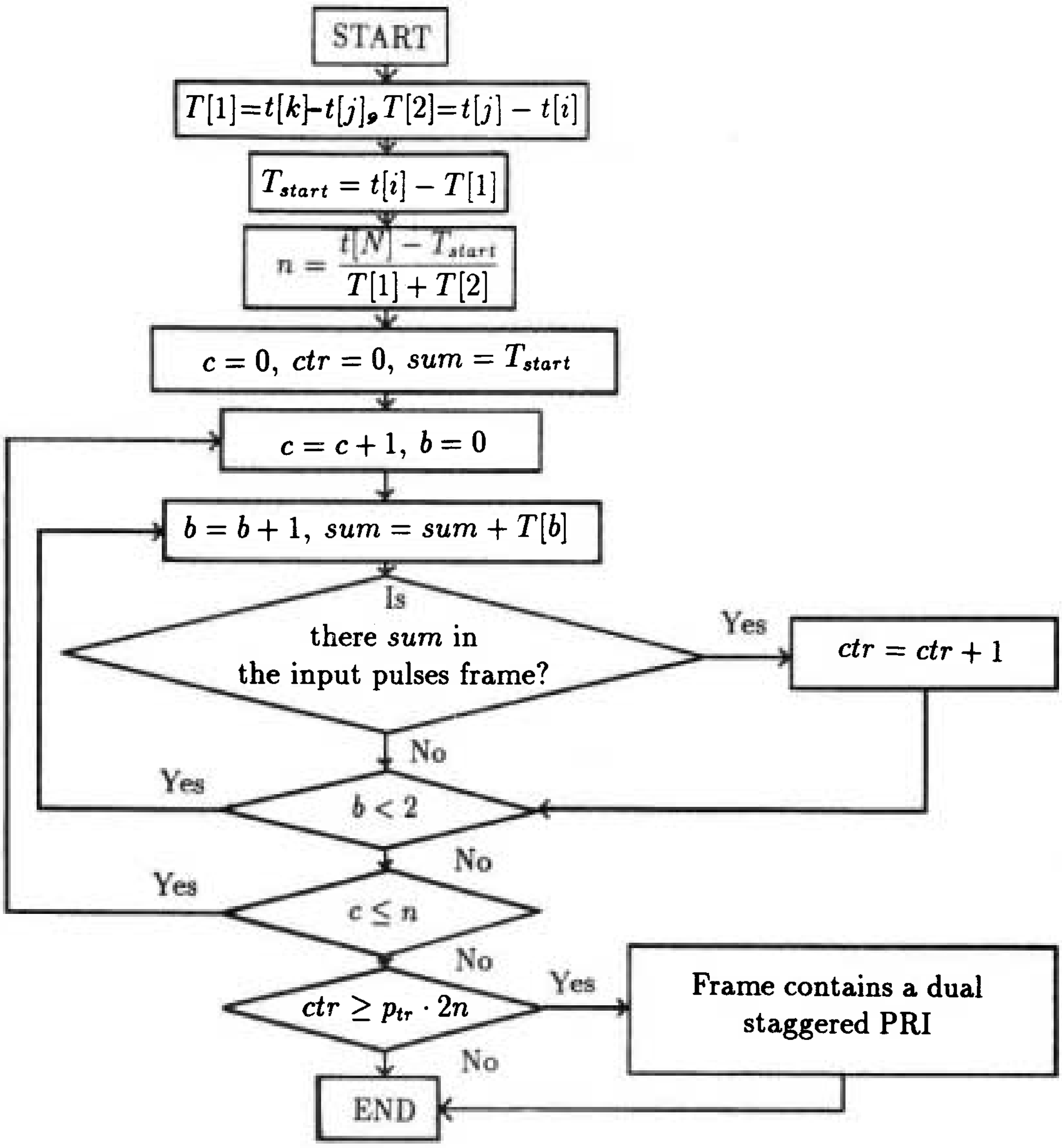

Figure 4. Algorithm for extracting staggered PRI pattern

$$T_{start} = t[i] - T[1] \tag{10}$$

The number of pulses of this emitter, n, is equal to

$$n = \frac{t[N] - T_{start}}{T[1] + T[2]} \tag{11}$$

For the staggered PRI of level three, the algorithm shown in Figure 4 has been used. This algorithm has been easily extended to identify staggered PRI of higher level. This can be done by considering $n+1$ received pulses as the first $n+1$ pulses of a staggered PRI of level n. It is also possible that there exist more than one staggered PRI pattern in the given frame. By changing those pulses that we have used to determine PRI patterns $(t[i], t[j], \ldots)$, all the possible staggered PRI patterns can be extracted.

4) *Pattern Extraction for the Dwell Switch PRI:*

Dwell switch PRIs can be identified by the parameters such as *T[1]* (the value of the first PRI), *n[1]* (the frequency of the first PRI), T[2] (the value of the second PRI), n[2] (the frequency of the second PRI). TOAs of the pulses in such a pattern can be written as $(T_{start} + T[1], \ldots, T_{start} + n[1]\,T[1], T_{start} + n[1]\,T[1] + T[2], \ldots, T_{strart} + n[1]\,T[1] + n[2]\,T[2], \ldots)$. Therefore, if we consider *t[i]* and *t[j]* as the first and second pulses for the *T[1]*, and *t[k]* and *t[l]* as the first and the second pulses for the *T[2]*, we can use the algorithm shown in Figure 4 to determine the dwell switch PRI patterns.

II. EMITTER IDENTIFICATION BASED ON PATTERN COMBINATION

In this section, different extracted patterns, which have been obtained based on the algorithm described in the previous section, are combined to identify possible emiters. Different combinations have been used. TOAs of all pulses located in these combinations have been computed and compared with the real TOAs of the pulses in the frame. Whenever one of these combinations has the same TOAs as the real pulses in the frame, up to P_{tr}, the patterns of that combination are considered as the identified emitters. The possibility of that combination is also given. All the information about these emitters have been saved in a data bank.

In order to reduce computation time, the operations for pattern extraction, and pattern combination have been performed on the first frame. For the next frame, TOAs of the pulses in the frame have been compared with TOAs of the emitters saved in the data bank. If these are the same, up to P_{tr}, the saved information has been used for the next frame. Otherwise, patterns extraction and patterns combination will be performed and the information in the data bank will be updated.

For the simple cases, like the environment with one constant PRI or jitter PRI, without extracting all possible patterns, emitter identification has been done first. This makes the algorithm more efficient. Also, some of the patterns which cannot combine in order to produce a combination are considered separately.

III. SIMULATION RESULTS

On the basis of the algorithm described in section I, a computer program has been designed and different examples have been applied to this algorithm. The results show that the algorithm is highly effective.

As an example, a complex environment including 19 emitters is illustrated in Table 4. It should be mentioned that the emitters are considered to belong to the four clusters. However, the nominal value of these emitters is the mean of the data created by a program which produces Gaussian random variables with variances σ_f, σ_{DOA}, and σ_{TOA} [7] for the frequency, DOA, and TOA, respectively. 3 percent of the generated data have been deleted randomly.

A computer program has been used to sort the pulses of these emitters according to their TOA. Then, the pulses have been applied to the computer program for the deinterleaving. P_{tr} is considered to be .97 according to x= 3 in Equation 7. Any number less than .97 gives the same result, but the process time increases. The results of neural network clustering based on DOA and RF are shown in Table 5. As expected, 4 clusters have been specified with the parameters within the defined variance of nominal values. In this table, the number of the assigned pulses to each cluster is given.

The assigned pulses to each cluster have been applied to the algorithm for the deinterleaving based on TOA. First, all possible constant PRI patterns are extracted. Then all possible jitter PRI patterns, staggered PRI patterns, and dwell witch PRI patterns are extracted separately. Pattern combination has been performed and the emitters are identified. Table 6 shows the simulation results for the 5 random

TABLE 4: An Example for a Complex Environment, Frame Time =50 msec, no. of Emitters= 19, σ_f= .25GHz, σ_{DOA}= 2.5°, and σ_{TOA}= 5ns, T_{start}= 0.0ms

Source No.	Frequency (GHz)	DOA (Degree)	PRI	PRI Parameter
1	8.2 GHz	25	Constant	T= 1.682 msce,
2	8.2 GHz	25	Constant	T= 1.83 msec,
3	8.2 GHz	25	Jitter	T= 2.0 msec, Tol= 0.05 msec
4	8.2 GHz	25	Staggered	T_1= 1.24 msec, T_2= 1.63 msec,
5	8.2 GHz	25	Staggered	T_1= 0.875 msec, T_2= 1.92 msec, T_3= 1.69 msec,
6	9.5 GHz	42	Contant	T= 1.56 msec,
7	9.5 GHz	42	Constant	T= 2.18 msec,
8	9.5 GHz	42	Staggered	T_1= 1.87 msec T_2= 1.20 msec
9	9.5 GHz	42	Staggered	T_1= 1.38 msec, T_2= 1.56 msec, T_3= 2.12 msec,
10	9.5 GHz	42	Dwell switch	T_1= 1.65 msec, T_2= 1.43 msec, n_1= 3, n_2= 4,
11	12 GHz	11	Jitter	T= 2.2 msec, Tol= 0.05 msec
12	12 GHz	11	Staggered	T_1= 1.482 msec, T_2= 1.795 msec, T_3= 1.706 msec,
13	12 GHz	11	Staggered	T_1= 1.82 msec, T_2= 0.94 msec, T_3= 1.36 msec,
14	12 GHz	11	Dwell switch	T_1= 1.23 msec, T_2= 0.652 msec, n_1= 2, n_2= 3,
15	12 GHz	11	Dwell switch	T_1= 1.15 msec, T_2= 2.25 msec, n_1= 2, n_2= 3,
16	10.7 GHz	56	Constant	T= 0.875 msec,
17	10.7 GHz	56	Staggered	T_1= 1.38 msec, T_2= 1.524 msec,
18	10.7 GHz	56	Staggered	T_1= 1.28 msec, T_2= 1.62 msec, T_3= 1.47 msec,
19	10.7 GHz	56	Dwell switch	T_1= 1.45 msec, T_2= 1.75 msec, n_1= 2, n_2= 4,

selected environments with the successfully detected emitters. The simulation results show the full effectiveness of the algorithm. During the algorithm test, the number of false detected emitters was zero.

TABLE 5: The Results of Neural Network Clustering Based on DOA and RF of the Example of Tables 3 and 4.

	Frequency (GHz) of the cluster center	DOA (Degree) of the cluster center	Number of the assigned pulses
Cluster No. 1	9.50 GHz	41.9°	147
Cluster No. 2	8.20 GHz	25.2°	147
Cluster No. 3	10.7 GHz	56.0°	155
Cluster No. 4	12.0 GHz	11.0°	172

TABLE 6: The Results of Computer Simulation for 5 Different Environments with the Number of False Identifications

	Detected emitters with success				False emitters			
Case	Cons.	Jitt	Stag.	Dwel.	Cons.	Jitt	Stag.	Dwel.
1	1	1	0	0	0	0	0	0
2	1	0	2	2	0	0	0	0
3	3	1	4	4	0	0	0	0
4	5	2	3	2	0	0	0	0
5	5	3	6	5	0	0	0	0

The case 5 in Table 6 is the simulation results of the example presented in Table 4.

It should be noted that the algorithm has been tested for more than 100 environments. The emitter identification has been performed without any error emitter identification.

For the comparison between this algorithm and that of using TOA difference histogram [3], the example of Table 4 is applied to both algorithms. Up to tenth order TOA difference are computed. However, the results of TOA difference histogram were ambiguous and the identification was impossible. Detailed evaluation of a practical implementation of this algorithm is required with actual scenario.

CONCLUSION

New algorithm for high-accuracy deinterleaving of radar pulses has been presented which can perform in radar environments with complex signal types. Good features of this algorithm are its effectiveness in TOA deinterleaving and its ability to identify different groups of possible radar with the degree of uncertainty. The contribution of various errors such as measurement errors and missing pulses are considered in the simulation of the input data. The algorithm can easily be implemented using parallel processing to give a substantial increase in performance.

REFERENCES

1. C. A. Hanna, "The Associative Comparator: Adds New Capabilities to ESM Signal Processing," Defense Electronic (February 1984), 51-62.
2. D. J. Milojevic and B. M. Popovic, "Improved Algorithm for the Deinterleaving of Radar Pulses," IEE Proc. F, Commun., Radar & Signal Process., Vol. 139, No. 1, (1992), 98-104.
3. J. A. V. Roger, "ESM Processor System for High Pulse Density Radar Environments," IEE Proc. F, Commun., Radar & Signal Process., Vol. 132, No. 7, (1985), 621-625.
4. R. Beale and T. Jackson, "Neural Computing: An Introduction," Adam Hilger, (1990).
5. J. E. Dayhoff, "Neural Network Architecture," Van Nostrand Reinhold, (1990).
6. D. R. Wilkinson & et al. "Use of Metric Techniques in ESM Data Processing," IEE Proc. F, Commun., Radar & Signal Process., Vol. 132, No. 4, (1985).
7. R. G. Wiley, "Electronic Intelligence: the Analysis of Radar Signals," Artech House, (1982).