Improved algorithm for the deinterleaving of radar pulses

D.J. Milojević
B.M. Popović

Indexing terms: Algorithms, Radar and radio navigation, Signal processing

Abstract: The paper presents an improved method for the deinterleaving of radar signals, based on a time of arrival analysis and the use of the sequential difference histogram (SDIF) for determining the pulse repetition interval (PRI). The optimal detection threshold in the SDIF histogram is derived, which greatly contributes to the efficiency of the algorithm. The algorithm is applied to classic, frequency-agile and staggered PRI radar signals. It is shown that the new method is very successful in high-pulse-density radar environments and for complex signal types. Special attention is given to an application of this method to the multiple-parameter deinterleaving algorithm.

1 Introduction

Deinterleaving of radar pulses, as an important part of a radar reconnaissance system, is a process of detection and recognition of different, simultaneously active, radar emitters. It assumes that received pulses have to be sorted, depending on their specific radar emitters.

Deinterleaving algorithms are based on the analysis of various parameters of the received radar pulses, such as time of arrival, angle of arrival, pulse amplitude, pulse width and carrier frequency. The algorithm for deinterleaving presented in this paper belongs to the family of so-called 'interval-only' algorithms. It uses only the information about the time of arrival (TOA) for deinterleaving.

The original version of this algorithm [1] is based on the so-called CDIF (cumulative difference) histogram of TOA pulses and is described in the following Section. A modified and improved version of this algorithm, based on the new sequential difference histogram technique (SDIF), is described here.

2 Basic algorithm description

The basic algorithm [1] leans on the CDIF histogram analysis. The CDIF histogram is formed by using the TOA differences; the calculated TOA difference between adjacent pulses (distance 1) is called the first difference. The TOA difference between each pulse and the next pulse but one (distance 2) is the second difference, and so on. The CDIF histogram assumes the accumulation of histogram values for each difference level. Potential values of the pulse repetition interval (PRI) will correspond to the histogram peaks.

For every TOA difference of level c, the CDIF histogram is formed and the new threshold is defined (the choice of the threshold will be discussed later). Each histogram value, and double that value, is compared to the threshold, and if none of these couples exceeds the threshold, the next TOA difference (of level $c + 1$) and the new cumulative histogram will be calculated.

If the potential PRI is identified, the algorithm performs the *sequence search* [1], i.e. it looks for a group of pulses that form a periodical pulse train, with periods equal to PRI. Such a group of pulses is known as a PRI sequence. If the search is successful, the PRI sequence will be extracted from the input buffer and the new CDIF histogram will be formed (starting from the first difference). This process is repeated as long as there are enough pulses in the input buffer to form any PRI sequence. If there is no detection (i.e. pulses do not form the PRI sequence), the next difference will be calculated. If none of the histogram values exceeds the threshold, the next difference will be calculated as well. If more than one histogram value exceeds the threshold, the sequence search is performed for every potential PRI value, starting from the lowest. The technique presented in Reference 1 is less sensitive to interference and missing pulses than conventional published techniques, such as the delta τ-histogram, which is formed of all TOA differences [2].

Fig. 1 shows the four successive CDIF TOA histograms of a signal which can represent either two interleaved stable sequences of pulses with equal PRI, or a two-level staggered signal with stagger frame rate equal to the PRI. By this algorithm, a staggered signal of level N will be identified as N signals with a constant PRI equal to the stagger frame rate [1]. To extract the true PRI value, the difference level must be increased to four.

The most significant drawback of the original algorithm, in our opinion, concerns the high number of difference levels required even in very simple cases, such as that in Fig. 1. By requiring the second harmonics to be present, the PRI calculation is limited to cases where the sequences of three events occur, rather than only the pairs. However, the same thing can be achieved in the second part of the algorithm, in the sequence search, using the appropriate conditions for the detection and extraction of PRI sequences. Therefore, if we discard the condition that the second harmonics have to be present, the accumulation in the difference histograms is no longer necessary. This was our main reason for using the sequential difference histogram, which does not include accumulation in the successive difference levels.

Paper 8413F (E5, E15), first received 2nd November 1990 and in revised form 28th May 1991

The authors are with the Institute of Microwave Techniques and Electronics, Bulevar Lenjina 165b, 11071 Novi Beograd, Yugoslavia

The key point in any histogram detection scheme is the threshold function. For the CDIF histogram, it can be proved (Section 4) that the optimal threshold is

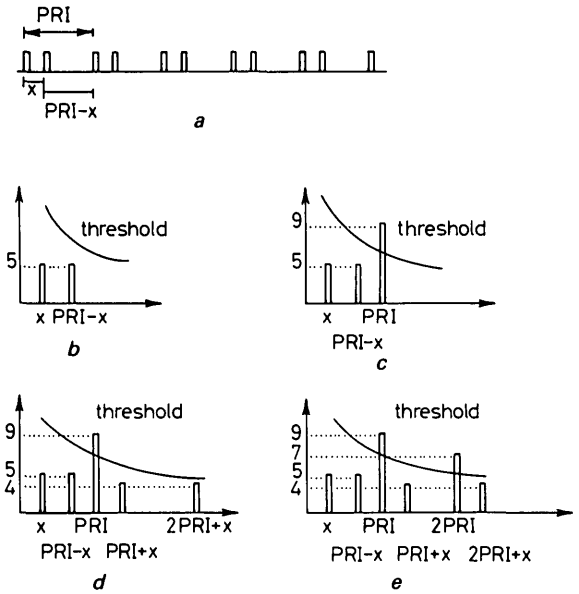

Fig. 1 *CDIF TOA histogram of two interleaved radar signals*

a Pulse train of two interleaved radar signals with the same PRI (or two-level staggered radar with stagger frame equal PRI)
b First difference
c Second difference
d Third difference
e Fourth difference

inversely proportional to the bin ordinal number. However, for the SDIF histogram it will be proved that the optimal threshold function is some form of the exponential function.

Computer simulations have shown another imperfection in the original algorithm. In cases where a great many pulses are missing, the harmonics (multiples of the true PRI value [1]) have been detected in the CDIF histogram, instead of the true PRI values. Therefore, the sequence search has extracted the false PRI sequences corresponding to the harmonics of the true PRI value.

Unintentional PRI variations due to transmitter circuit imperfections can cause serious problems in the algorithm, because the histogram peaks decrease (if the total number of received pulses is constant) and could not exceed the threshold. Even if the histogram value corresponding to the true PRI value exceeds the threshold, greater tolerances are necessary in the sequence search for the detection of corresponding PRI sequences. However, this can cause detection and extraction of an incorrect PRI sequence, because interleaved pulses of other emitters could be mistakenly included in a PRI sequences.

3 Improved algorithm description

The new algorithm generally consists of two parts, namely the estimation of the PRI and the sequence search. The estimation of the PRI, as the key part of the deinterleaving algorithm, will be described later. The sequence search is also a very important part of the algorithm, allowing many improvements in speed, reliability and efficiency to be achieved. However, the sequence search we used in the new algorithm is the same as in Reference 1. The main reason for this is to allow a fair comparison between the original and the new, improved algorithm.

Fig. 2 shows the SDIF histogram of two interleaved radar signals with the same PRI from Fig. 1. In the SDIF histogram only current differences exist, and the histogram is much clearer than the corresponding CDIF histogram. To extract the true PRI, it is sufficient to

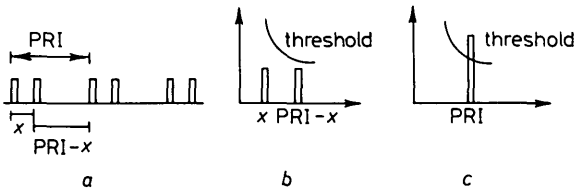

Fig. 2 *SDIF histograms of two interleaved radars with the same PRI*

a Pulse strain *b* First difference *c* Second difference

compute the second difference and to compare the histogram value with the threshold. There is no need to compare the double PRI with the threshold, hence the computation time is more than halved.

The other imperfections of the basic algorithm could be removed or reduced by the following modifications:

(i) When the first difference is computed, if only one histogram value exceeds the threshold, that value becomes the potential PRI for which a sequence search will be performed. However, if many emitters are present, the first difference histogram will produce a few values exceeding the threshold, none of which corresponds to the right PRI value. That is why the next difference has to be calculated without the previous sequence search (the basic algorithm suggested in Reference 1 did not foresee this step). Fig. 3 shows the need to skip the first

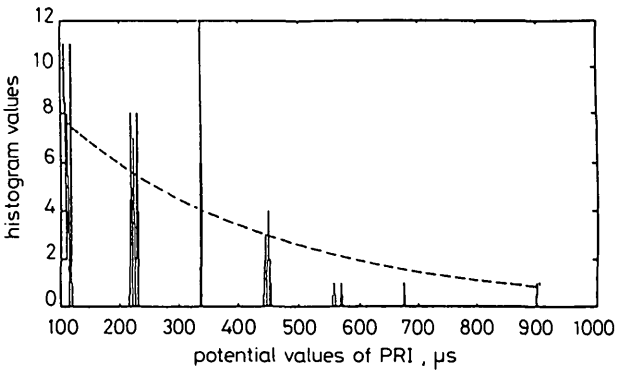

Fig. 3 *SDIF histogram of the TOA first difference in a complex radar environment*

difference in the case of three interleaved radar signals: two classic radars (with PRIs of 217 and 248) and one stagger (with a stagger frame rate of 318). The first difference in the SDIF histogram does not present the true PRI values, because they are hidden in higher differences.

(ii) In a real situation, the random jitter of PRI values is always present, owing to the imperfections of the transmitter circuits. Because of this, similar PRI values in the SDIF histogram, grouped around the true PRI value, can produce histogram values exceeding the threshold. If the range of PRI values is no greater than the permitted tolerances, the sequence search is performed for the central value, which represents the potential jittered PRI. Fig. 4 illustrates this case: the histogram values exceeding the threshold correspond to one detected emitter with a random jitter PRI of central value 263.

(iii) In cases where a great number of pulses are missing, the harmonics of the basic PRI can produce significant components in the SDIF histogram. These values could be so dominant as to exceed the threshold. This is

not a problem if the histogram value corresponding to the true PRI value also exceeds the threshold, because the PRI analysis and the sequence search start from the

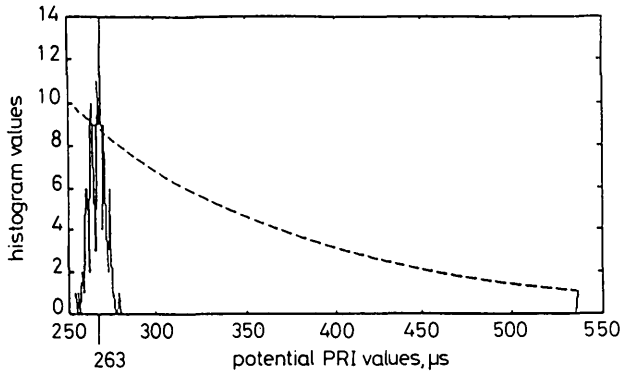

Fig. 4 *SDIF histogram of random jittered PRI signal*

lowest PRI having a histogram value that exceeds the threshold. But if the true PRI does not exceed the threshold, its harmonics will be used in the sequence search, and incorrect PRI sequences could be extracted. Fig. 5 shows the SDIF histogram with a true PRI value of 428 under the threshold, and one harmonic of 856 exceeding the threshold.

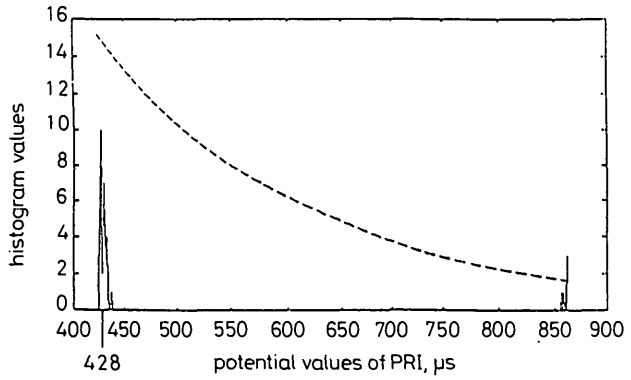

Fig. 5 *SDIF histogram of TOA for many missed pulses*

To avoid 'false' detection, subharmonic checking is necessary: the histogram maximum is found and if it does not exceed the threshold, the first (lowest) value exceeding the threshold is checked. If the corresponding PRI value presents some harmonic of the PRI value corresponding to the histogram maximum, this then becomes the potential PRI for which the sequence search is performed.

If the lowest PRI value which exceeds the threshold is not a multiple of the histogram maximum PRI value, the sequence search is performed for all PRI values which exceed the threshold, starting with the lowest. This analysis of harmonics and subharmonics also represents a modification of the basic algorithm [1].

(iv) The choice of threshold is most important for reliable and fast emitter detection, using the SDIF histogram. The threshold must follow the distribution function of the process to prevent the detection of many 'false' emitters. In the following Section, the optimal detection threshold function is derived.

The flowchart of the improved algorithm is shown in Fig. 6. For every difference of level c, the SDIF histogram is formed, the threshold is calculated and all histogram values exceeding the threshold after the subharmonic check become the potential PRI for sequence search. If the extraction of PRI sequences is successful, the process is repeated until the extraction of pulse trains, or until five pulses remain in the entry buffer. If the sequence search cannot extract a PRI sequences, the next difference is calculated, the new threshold is set and the whole process is repeated. At the end of the PRI analyses, the stagger recognition is performed.

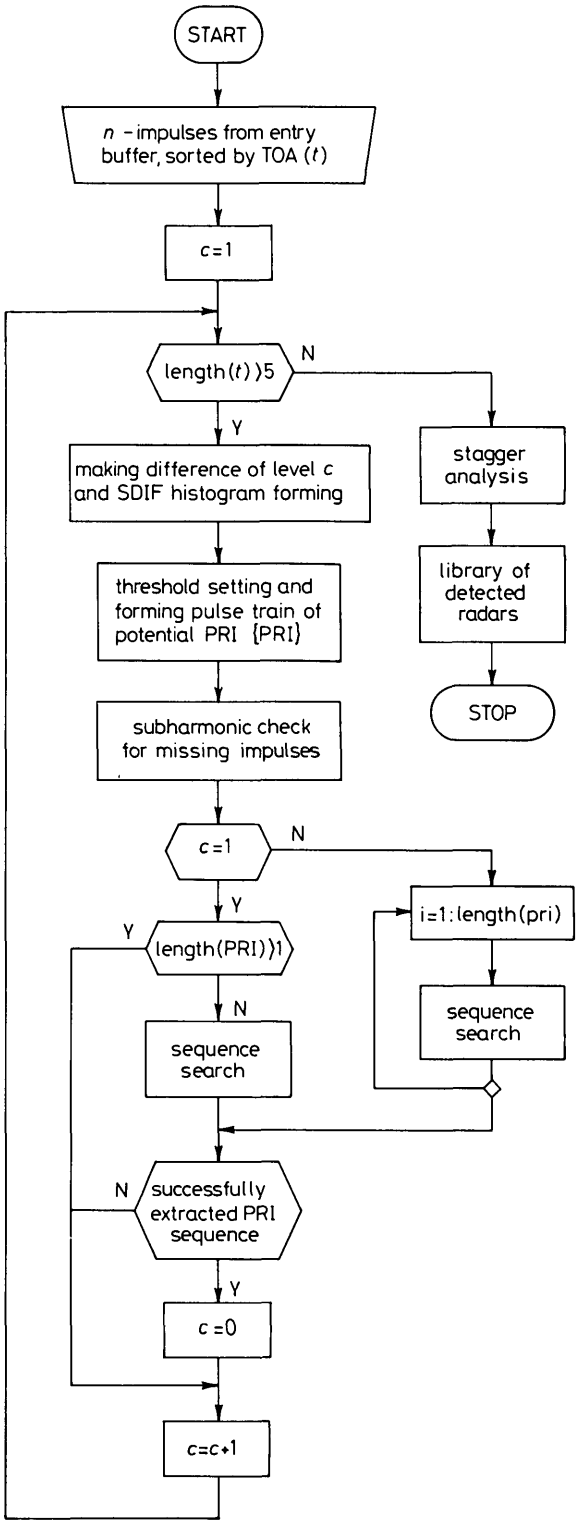

Fig. 6 *Flowchart of modified 'interval-only' algorithm based on SDIF histogram of TOA*

4 Optimal detection threshold for SDIF histogram

To determine which histogram peak corresponds to the potential pulse repetition interval, it is necessary to find the optimal threshold function, depending on the bin number in the SDIF histogram.

It is easy to see that a threshold value is inversely proportional to the bin number τ. As the bins in the histogram correspond to the intervals between pulses, and as observing time is bounded, it follows that the larger the observed time interval between pulses, the smaller the number of appearances of that interval in a finite observation time. Mathematically, such a threshold can be represented as

$$p(\tau) = \frac{xE}{\tau} \tag{1}$$

where E is the total number of observed pulses and x is a constant less than 1.

If the number of observed pulses is large enough, and if several emitters are active at the same time, an interval between adjacent pulses could be considered as a random event, i.e. leading edge of pulses could be observed as *random Poisson points* [3], which a finite observed interval T divides at n subintervals. As we know, the probability that k of these n random placed points will lie in an interval $(t_1 - t_2)$ of length $\tau = t_2 - t_1$ is given by the poisson distribution [3]:

$$p_k(\tau) = \frac{(\lambda\tau)^k}{k!} e^{-\lambda\tau} \quad k = 0, 1, 2, \ldots \tag{2}$$

where eqn. 2 is derived from the assumption that $\lambda = n/T$, and that n and T increase indefinitely, but the ratio λ remains constant.

In eqn. 2, λ is a parameter of the Poisson process, and represents the average number of events in a time unit. As we observe pulse train from the different emitters active at the same time, the number of received pulses will be proportional to the total observation time, i.e. the total number of bins N.

The probability that one of the Poisson points lies in the interval τ can be derived from the previous expression for $k = 0$:

$$p_0(\tau) = e^{-\lambda\tau} \tag{3}$$

which is proportional to the probability that the interval between adjacent pulses has the value τ. In other words, the histogram of the first differences will have the form of eqn. 3. As the histogram is the estimate of the probability distribution function of a random event, the histogram of higher-level differences will also be in exponential form; the maximum histogram peak would be decreased due to the difference level c, because the number of pulses forming the histogram on level c is equal $E - c$. As the observed time interval is proportional to the number of samples, i.e. to the total number of bins in the histogram, the parameter of the Poisson process is $\lambda = (1/kN)$, where k is a positive constant less than 1.

From the previous discussion we can conclude that the optimal function of the detection threshold should take the following form:

$$\text{threshold } (\tau) = x(E - c)e^{-\tau/(kN)} \tag{4}$$

where E is the total number of pulses, N is the total number of bins in the histogram, and c is the difference level. The optimum values of the constants x and k are experimentally determined. The assumed uniform probability distribution for the occurrence of missing pulses is, in fact, superimposed on the Poisson distribution, and so the nature of the process is not changed and the threshold form follows the histogram peaks. Consequently, the constant x depends on the supposed maximum percentage of missed pulses.

Table 1 shows the simulation results of the deinterleaving algorithm for different thresholds. The algorithm is simulated on a PC386 computer, using the very high-level programming language MATLAB, and the processor times are expressed in flops (floating point operations). Emitters which are produced by the deinterleaving algorithm, but which do not really exist in the radar environment, are called 'false' emitters.

Table 1: Results of computer simulation

Threshold function	Number of successfully detected emitters			Number of 'false' emitters			Processor time [flops]		
	I	II	III	I	II	III	I	II	III
exp $(-\tau)$	5	7	3	0	1	0	285 K	341 K	188 K
$1/\tau$	2	3	2	5	10	2	823 K	1.32 M	514 K
$1/\sqrt{(\tau)}$	4	5	3	2	5	1	489 K	827 K	333 K

I = 5 radars (2 stagger, 2 frequency-agile, 1 classic)
II = 7 radars (3 stagger, 2 freqency-agile, 2 classic)
III = 3 radars (3 stagger and frequency-agile)

It is obvious that the new exponential threshold gives much better results, because an exponential function best corresponds to the nature of the process.

Fig. 7 shows the different forms of threshold function used in the simulation. It can be seen that the new threshold closely follows the histogram peaks.

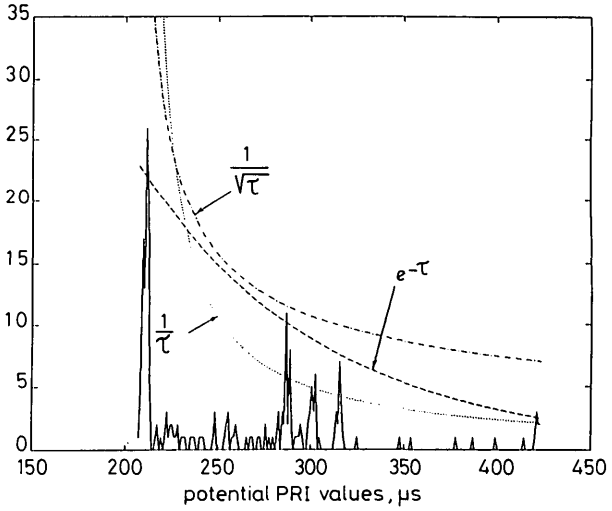

Fig. 7 *Different forms of threshold in SDIF histogram*

In Reference 1, the threshold function was not discussed but only shown in Figs. 5 and 6 [1]. The values in the CDIF histogram [1] are accumulated from previous differences, and therefore the decision threshold would be a summation of exponential functions, which at the limit should tend to a $1/\tau$ relationship given by eqn. 1. This confirms the optimality of the exponential threshold.

With the new threshold, the time of analysis is highly reduced because the improved algorithm gives only a few false PRI values for which the sequence search should be performed. It can also be seen that, with our choice of exponential threshold, only the low-level differences have to be calculated, reducing the number of histograms. This is another reason why the new algorithm is much faster than the old one [1].

5 Multiple-parameter deinterleaving algorithm

The improved PRI analysis shown in the preceding Sections could be successfully applied as a part of a multiple-

parameter deinterleaving algorithm [4, 5, 6, 7], where the PRI analysis would be performed on groups of pulses previously sorted by azimuth (DOA), frequency (RF), pulse width (PW), or other parameters. By using other pulse parameters besides the time of arrival, new possibilities for efficient deinterleaving are opened up.

In Reference 6, the deinterleaving process takes place in two stages: the DOA RF filter splits each entry block of pulses into a number of pulse batches, which are then processed sequentially by the TOA difference histogram and split into individual pulse chains. The proposed DOA RF filter, as a two-dimensional algorithm, sorts the pulses into DOA buckets and then uses large gaps in the RF values to define clusters. In the case of frequency-agile radars, the main issue is the optimal choice of the distance threshold, which can avoid fragmenting pulses from one radar into different groups, or forming excessively large groups. Metric technique plays an important role in DOA RF filtering.

The two other multiple-parameter algorithms, proposed in Reference 7, also use a metric technique for multidimensional pulse clustering, which is weighted to reflect the resolving ability of each parameter. The TOA difference histogram is then calculated on these clusters. In these algorithms, the major challenge is the separation of stable and agile radars, as well as the correct grouping of radars with overlapping characteristics.

The main reason for considering the multiple-parameter deinterleaving algorithms was not to discover the new, more efficient algorithm, but rather to see the influence of new pulse parameters included in the deinterleaving process, and to compare the speed and reliability of such algorithms with the improved TOA deinterleaving algorithm described in Section 3.

The analysis of other pulse parameters in the new algorithm is predominantly performed by using histograms. Azimuth is chosen as the most important deinterleaving parameter, because of its stability and reliability [4], and the azimuth histogram is formed from all the pulses in the input buffer. The pulses grouped about every histogram maximum correspond to the different emitters, and it is necessary to separate these pulses according to their azimuth values.

Using histograms presents the problem of determining the boundaries of histogram groups. One possible solution is that the local minima of the averaged histogram values are declared as group boundaries. Then, pulses from the input buffer with the same, or similar, azimuth are separated, and sorted into azimuth groups. Inside every azimuth group, it is necessary to analyse the next parameter.

At first sight it seems natural to choose frequency as the second deinterleaving parameter [6]. In that way we could have pulse clustering in the azimuth frequency space. In the case of frequency-agile radars, frequency clustering can cause the appearance of a large number of pulse groups, and, consequently, 'false' emitters. By clasifying the pulses into frequency groups inside one azimuth group, the correlation between pulses from the same emitter could be lost, and so the application of PRI analysis could be almost impossible.

As is well known, the pulse width and the amplitude are unreliable parameters, and also inconvenient for deinterleaving [4], and so the time of arrival is chosen for the second deinterleaving parameter. The new improved algorithm for PRI analysis is used for fine pulse clustering inside every detected azimuth group.

As a result of PRI analysis, inside each azimuth group, pulses with the same or similar PRI values are sorted into PRI groups. For better classification of detected emitters, the carrier frequency is taken as the third parameter for deinterleaving, which means that inside each PRI group the histogram of frequency is formed and analysed.

For radars with fixed carrier frequencies, the frequency histogram will produce a single significant peak. However, if multiple significant peaks appear in the frequency histogram, we can conclude that the emitter is frequency-agile. Frequency clustering is not carried out, and the frequency histogram only tells us whether the pulses from the PRI group belong to a frequency-agile radar or not.

Stagger analysis is performed last. The extracted group of pulses, characterised with the same azimuth, the PRI and the frequency, will be assumed to correspond to the single detected emitter. The flowchart of this multiple-parameter algorithm is shown in Fig. 8.

An important improvement on the overall multiple-parameter algorithm can be achieved if we introduce the two-dimensional sequence search into the TOA deinterleaving algorithm. This consists in analysing the pulse width as well as the time of arrival, ensuring reliable PRI extraction even in the case of very high interleaved pulse density. Although the algorithm becomes more complicated, the speed of the process is not decreased; on the contrary, the gain in speed amounts to about 1 per cent.

It is interesting to see how much the multiple-parameter deinterleaving algorithm will improve reliability and speed, compared with the improved TOA algorithm described in Sections 3 and 4. When only the time of arrival is used for the deinterleaving, the SDIF histogram is formed from all the pulses in the input buffer, and the sequence search is performed on the entire buffer. The SDIF histogram technique gives very good results and high accuracy, because all received pulses are available for the analysis. However, in the case of high pulse density and a large number of interleaved signals, the PRI analysis by SDIF histogram becomes complicated, and the detection of emitters is no longer reliable.

By the azimuth analysis, the input buffer is divided into smaller pulse groups and each group is analysed by the SDIF histogram technique, but forming the SDIF histogram is now much faster. Because of the small number of pulses, the sequence search is also faster. In the case of complex radar environments and a great number of pulses, azimuth clustering becomes necessary not only because of important gains in speed, but also the reliability of emitter classification.

Table 2 shows some results of computer simulations using MATLAB. Two methods for deinterleaving are compared: the interval-only algorithm (where only the time of arrival is analysed) and the multiple-parameter algoritm (where azimuth (DOA), time of arrival (TOA) and carrier frequency (RF) are the parameters used).

Experience from many algorithm tests has confirmed that azimuth is the best parameter for the initial clustering. Forming azimuth groups makes the largest contribution to the increase of algorithm speed. From Table 2, the gain in speed is seen to be about 70% for the example *I* (two classic radars) to approximately 300% for the example *V* (ten radars: three classic, four stagger, three frequency-agile). We can conclude that the more complex the radar environments, the greater the gain in speed that could be obtained.

In the case of small numbers of interleaved radars, the azimuth clustering has little influence on the reliability of

emitter classification. However, in a complex radar environment, the reliability is still increased, as can be seen from the last example (10 radars) in Table 2. The interval-only algorithm has successfully detected seven emitters and ten 'false' emitters, but the multiple-parameter algorithm has successfully detected nine emitters and three 'false' emitters.

From this last example, it may be seen that the new algorithm gives slightly poorer results than in previous examples. The reason is the sequence search algorithm

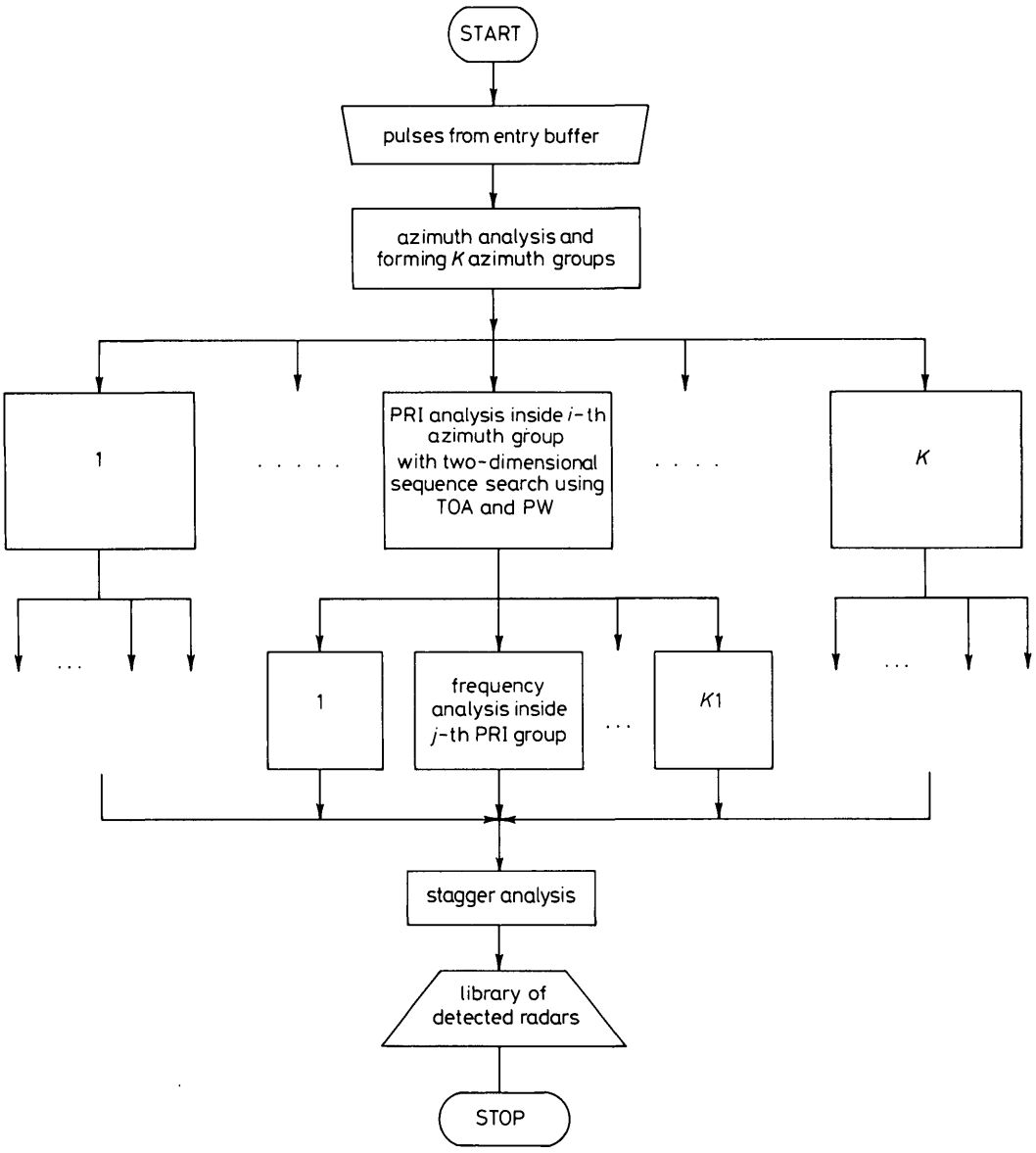

Fig. 8 *Flowchart of multiple-parameter deinterleaving algorithm*

Table 2: Results of computer simulation

	Parameters used in analysis	Successfully detected emitters			'False' emitters			Processor time [flops]
		clas.	stag.	agile	clas.	stag.	agile	
I	DOA, TOA, RF	2	0	0	0	0	0	56 271
	only TOA	2	0	0	1	0	0	73 432
II	DOA, TOA, RF	1	1	2	0	0	1	137 855
	only TOA	1	1	2	0	0	0	292 398
III	DOA, TOA, RF	1	3	0	0	1	0	326 212
	only TOA	1	2	0	2	1	0	547 892
IV	DOA, TOA, RF	2	2	1	0	0	0	282 097
	only TOA	1	0	1	5	3	0	889 325
V	DOA, TOA, RF	3	3	3	1	2	0	623 417
	only TOA	2	3	2	6	2	2	1 921 313

I = 2 radars (2 classic)
II = 4 radars (1 classic, 1 stagger, 2 frequency-agile)
III = 4 radars (1 classic, 3 stagger)
IV = 5 radars (2 classic, 2 stagger, 1 frequency-agile)
V = 10 radars (3 classic, 4 stagger, 3 frequency-agile)

within the TOA deinterleaving algorithm, which for very high pulse density extracts the pulses belonging to different emitters, with close times of arrival. In our simulation we used the one-dimensional sequence search algorithm from Reference 1. Better results can be obtained if we use the two-dimensional sequence search algorithm based on the time of arrival and pulse width. The main role of the pulse width as the second parameter in the sequence search, is to prevent the extraction of the wrong pulses into the PRI sequence, except in the case of pulses with very close pulse width values.

Table 3: Results of computer simulations

Sequence search algorithm	Parameters used in deinterleaving	Successfully detected emitters			'False' emitters			Processor time [flops]
		clas.	stag.	agile	clas.	stag.	agile	
1-D	DOA, TOA, RF	3	3	3	4	1	0	679939
2-D	DOA, TOA, PW, RF	3	3	3	1	1	0	674065

The comparison of one- and two-dimensional sequence search algorithms in the case of ten radars: three classic, four stagger (of levels 2, 3, 4, 5) and three frequency-agile.

Table 3 shows the comparison of two sequence search algorithms (one-dimensional and two-dimensional) implemented in the multiple-parameter algorithm described earlier. The observed radar environment consists of ten radars (there are 934 interleaved pulses in the entry buffer): three classic, three frequency-agile and four staggered radars of levels 2, 3, 4 and 5. As the result of the deinterleaving, we obtain nine successfully detected emitters in both cases, and five 'false' emitters with the one-dimensional sequence search. The false emitters are generated from the five-level stagger radar, which is not detected. (This environment is more complicated than the environment from the last example from Table 2, because the higher pulse density and stagger level are used). With the two-dimensional sequence search algorithm, two 'false' emitters are detected. The increasing reliability and process speed can be seen from table 3.

6 Conclusion

This paper presents a modified and improved algorithm for deinterleaving radar pulses, based on TOA analysis using the SDIF histogram and its application to a multiple-parameter deinterleaving algorithm.

Computer simulation shows that the presented algorithm based on TOA analysis produces very good results for classic, staggered and frequency-agile radars. More reliable and faster detection of emitters, especially in complex radar environments, can be achieved using the multiple-parameter deinterleaving algorithm, which includes the new SDIF histogram analysis of TOA. An additional improvement can be achieved if we include the two-dimensional sequence search algorithm within the TOA analysis, which uses pulse width as well as time of arrival.

The use of all available pulse parameters (DOA, TOA, PW, RF) and the employment of multidimensional algorithms in the process of detection and recognition of radar signals, offer great possibilities for efficient deinterleaving algorithms.

7 Acknowledgments

The authors are very grateful to the referees for their valuable comments.

8 References

1 MARDIA, H.K.: 'New techniques for the deinterleaving of repetitive sequences', *IEE Proc. F, Commun. Radar & Signal Process.*, 1989, **136**, (4), pp. 149–154
2 WILEY, R.G.: 'Electronic intelligence: the analysis of radar signals' (Artech House, 1982)
3 PAPOULIS, A.: 'Probability, random variables and stochastic processes' (McGraw Hill, 1986)
4 DAVIES, C.L., and HOLLANDS, P.: 'Automatic processing for ESM', *IEE Proc. F, Commun., Radar & Signal Process.*, 1982, **129**, (3), pp. 164–171
5 ROGERS, J.A.V.: 'ESM processor system for high pulse density radar environments', *IEE. Proc. Commun., Radar & Signal Process.*, 1985, **132**, (7), pp. 621–625
6 WILKINSON, D.R., and WATSON, A.W.: 'Use of metric techniques in ESM data processing', *IEE Proc. F, Commun., Radar & Signal Process.*, 1985, **132**, (4), pp. 27–36
7 MARDIA, H.K.: 'Adaptive multidimensional clustering for ESM'. IEE Colloquium on signal processing for ESM systems, 1988, Digest 81988/62, pp. 5/1–5/4