

# A Simple Way to Deinterleave Repetitive Pulse Sequences

YUJUN KUANG, QINGBO SHI, QIANBIN CHEN, LI YUN, KEPING LONG

Special Research Centre for Optical Internet and Wireless Information Networks (COIWIN),  
Chongqing University of Posts & Telecommunications, 400065, Chongqing, China

*Abstract:* This paper presents a simple but efficient algorithm for deinterleaving of periodical pulse sequences. Based upon time of arrival analysis, one or more pulse repetition intervals are determined from the sequential difference histogram just by detecting the maximum value in the histogram. The algorithm is compared to some typical signal sorting methods and implemented in C language program. It is shown that, the proposed algorithm is timesaving, efficient and easy to implement for real-time applications comparing with those based upon threshold functions or transform computing which is time-consuming.

*Keywords:* Periodical pulse deinterleave, TOA, SDIF, Radar and radio navigation, Signal processing

## 1 Introduction

In modern warfare, electronic supporting measures (ESM) and electronic counter counter-measures (ECCM) encounter increasingly arduous tasks when electromagnetic environments become severer and more complex. One important task of ESM is to collect information about how many radars and/or frequency and hopping (FH) mobile communication sets are simultaneously active and to identify their criticalities. This is done by the preprocessor in the reconnaissance subsystem, where deinterleaving plays a great important role in the detection and recognition process including both interception of the pulses and identifying each distinct emitter.

Deinterleaving is a kind of clustering analysis, which clusters interweaved pulses intercepted by a scout or by other means, into distinct groups belonging to respective emitters, according to the pulses' features and a lot literature has shown interests in radar pulses deinterleaving. In [1] and [2], pulse parameter measurement and two types of deinterleaving algorithms of Cutlass ESM system are discussed as "pigeon hole" technique and "slicing". In [3], a pigeon-hole-like [1, 2] algorithm with PRI estimation technique using TOA difference histogram analysis has been approached. Based on [3], an improved algorithm with an exponential threshold function has been proposed in [4], which uses only sequential difference histogram (SDIF) to reduce processing time. In [5, 6], radon transform is introduced to estimate PRI and extract pulses via image processing, which, however, acts similarly to histogram and involves additional complex transformation. In [7, 8], Noone and SHYU et al introduce neural network to deinterleave radar signals which involve complex neural elements. This paper first evaluates the performance of deinterleaving algorithm proposed by D.J. Milojevic and B.M. Popovic [4], and then gives a simple but efficient PRI estimator which is based upon peak-value-detection in TOA-difference histogram.

In Section 2, we gives a summary of typical deinterleaver and a brief description of exponential

function based PRI estimator [4]. In Section 3, the proposed algorithm is given and performance evaluation is illustrated in Section 4. Section 5 concludes the paper.

## 2 System Model and Definitions

In typical ESM systems, other description information of the intercepted pulses are also obtained to form a series of pulse description words (PDW), e.g., angle of arrival (AOA) or direction of arrival (DOA), pulse amplitude (PA), carrier frequency (CF), pulse width (PW), modulation of pulse (MOP) and so on. Deinterleaving is performed by analyzing the PDWs of the mixed pulse sequence, i.e., on the analysis of the correlation of the pulses according to their features.

However, typically, only TOA and AOA are used in most deinterleaving algorithms, because, for example, PA is much related with channel status and not reliable for identifying emitters. Though AOA cannot be precisely measured with state-of-the-art devices, emitters are often relatively fixed to the detector during the detector's short scanning interval ( $T_{\max}$ ), thus it is a reliable parameter for clustering. It is apparent that TOA is very important to identify periodical emitters. Therefore, in [1] and [2], deinterleaving algorithms belonging to the family of so-called 'multi-parameter' algorithm are discussed, which use only AOA and TOA to deinterleave the mixture.

Fig.1 illustrates structure of typical deinterleavers [1-4]. As assumed, each emitter emits pulses at certain interval, or pulse repetition interval (PRI). All the pulses are mixed by the channel and intercepted by a scout's detector and sorted according to each pulse's time of arrival (TOA). Deinterleaving is performed in two steps: 1) AOA-clustering, which first clusters pulses from certain degree of arrival; and 2) TOA-deinterleaving. As the most important and kittle step, TOA-deinterleaving is further divided into two steps: a) PRI estimation and b) sequence searching, though they may be working overlapped in time, e.g., in classical heuristic techniques. In other words, the deinterleaver obtains a possible PRI and begins sequence searching

to seek pulses to form a pulse sequence matching the estimated PRI, i.e., a PRI-sequence. If searching succeeds, the deinterleaver extracts the matched pulses from the massive mixture. The deinterleaver repeats the two steps repeatedly until the remaining pulses cannot form any PRI sequence. Apparently, the more accurate is the estimated PRI, the more likely the sequence searching succeeds, thus, much less time will be wasted by failure searching.

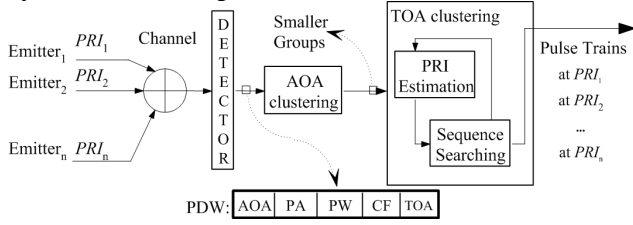


Fig.1 Pulse interception and deinterleaving

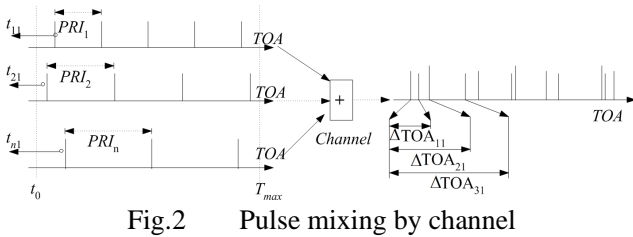


Fig.2 Pulse mixing by channel

### 3 SDIF and PRI Estimation

#### 3.1. SDIF description

It is natural to arrange the intercepted pulses according to TOA and to denote each pulse by its TOA when TOA-clustering is concerned (See Fig.2). Therefore, a pulse sequence of  $N$  pulses, in terms of TOA, can be expressed as:

$$\{t_i\} \triangleq \{t_1, t_2, \dots, t_{N-1}, t_N\} \quad (1)$$

where  $t_i$  denotes TOA of the  $i$ th pulse and  $t_j > t_i > 0, \forall j > i > 0$ , and

$$t_i \in \{t_{kl}\}, 1 \leq k \leq n, 1 \leq l \leq \lfloor T_{\max} / PRI_k \rfloor \quad (2)$$

where  $t_{kl}$  is the  $l$ th pulse of the  $k$ th emitter whose PRI is  $PRI_k$ , noting that when

$$t_{kl} = t_{k'l}, k \neq k' \text{ or } l \neq l' \quad (3)$$

occurs, a pulse missing may happen especially when their PDW.CF are close enough.

In [3, 4], TOA differences are calculated to produce sequential difference histogram (SDIF) for PRI estimation. TOA differences between adjacent pulses are said to belong to the first level difference ( $\{\Delta t_{i,j}\}$ ), and those between each pulse and the next but one belong to the second level difference ( $\{\Delta t_{2i}\}$ ), and so on (See Fig.2). In general, the sequential difference at level  $c$  is given as:

$$\{\Delta t_{ci}\} \triangleq \{\Delta t_{ci} | \Delta t_{ci} = t_{c+i} - t_i\}, \quad (4)$$

where  $\Delta t_{ci}$  denotes TOA difference and  $i = 1, 2, \dots, N - c, 0 < c < N$ .

All TOA differences of level  $c$  are calculated to form a SDIF [4] of level  $c$ . In contrast, when all TOA

differences at level  $c$  down to one are taken into account, the histogram is called cumulative difference histogram (CDIF) [3]. Apparently, CDIF involves more computation operations [4] than SDIF. Therefore, only SDIF is discussed in the following sections.

To illustrate, Fig.3-(a) shows a segment of a typical pulse sequence comprising three subsequences whose PRI are 12000, 15340 and 27000 denoted by pulse height of 1, 2 and 3, respectively, as well as the SDIF histogram of level 1 and level 2 shown in Fig.3-(b) and Fig.3-(c).

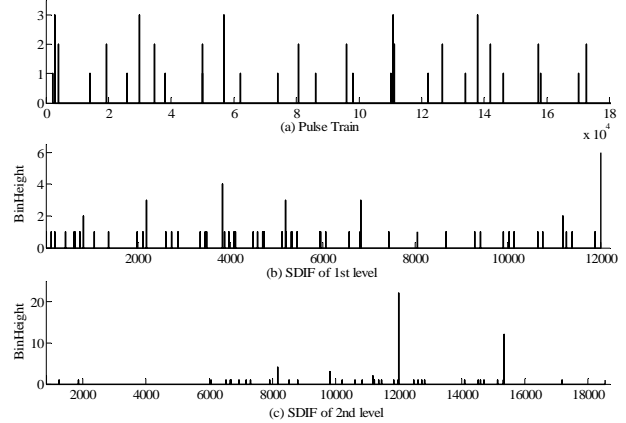


Fig.3 Illustration of TOA difference histogram

Note that the SDIF of 1<sup>st</sup> level gives a peak at 12000us, and in the SDIF of 2<sup>nd</sup> level, the bin-heights at 12000, 15340 are much higher than those elsewhere. In addition, further discussion will show that it is often true that the peaks in SDIF of certain level correspond to the right PRI.

#### 3.2. Identification of PRI

##### 3.2.1. Traditional PRI Identifications

When histogram is formed, the PRI identifying procedure begins. Because of its significance in TOA-deinterleaving, PRI estimation is always a hotspot. Most literatures presume that pulses' TOA can be regarded as random events characterized by certain probability distribution function, and then threshold functions are derived to obtain PRI(s) where bin-height(s) of SDIF of CDIF exceeds the thresholds [3, 4]. For example, in [4], on the supposition that leading edge of pulses (TOA) could be observed as random *Poisson* points, an optimal detection threshold function is derived as:

$$p(\tau) = x(E - c) e^{-\tau/kN}, \quad (5)$$

where  $E$  is the total number of pulses,  $N$  is the total number of bins in the histogram, and  $c$  is the difference level, while  $x$  and  $k$  are experientially determined constants. However, our MATLAB© simulation results show that the proposed algorithm in [4] maybe less efficient than the author believed and the experiential parameters  $x$  and  $k$  obtained from one scenario can not

be applied to others.

1) Evaluation of performances by process time in flops given by MATLAB is not exactly proportional to the real time that would be expended by real systems, especially in fixed-point systems\*.

2) In MATLAB, operations such as *sqrt*, *sin*, *cos*, *exp* and *log* take no flops if the operant is only a scalar, and they take only one flop for each item of X when X is a vector of size  $M \times N$ , where  $M \times N > 1$ .

3) More crucially, the experientially determined parameters ( $x, k$ ) are highly sensitive to the number of simultaneously active emitters and combination of their PRIs, which are usually not prior known to the ESM system. To illustrate, in Fig.4, threshold values together with SDIF are plotted under three threshold parameter ( $x, k$ ) pairs which are obtained from more than 50 simulations on five-radio cases. In Fig.4-(a) (five radios), we can identify three PRIs whose bin-height is above two of the thresholds, while in Fig.4-(b) (10 radios), no bin-height is above any of the thresholds. On the other hand, threshold parameters ( $x, k$ ) fit for 10-radio cases cannot be applied to five-radio cases either. Therefore, it is difficult to apply this algorithm to real ESM systems.

### 3.2.2. A Simple PRI Identification Approach

In Fig.4-(b) and 3-c, we have already seen that the TOA differences corresponding to extreme values in SDIF are correct PRIs. Other simulations prove that it is not a casual phenomenon, and it is suggested that the smaller the PRI is, the more often it makes a peak in low-level SDIF.

Here, we have recalled<sup>†</sup> the very simple and efficient technique to obtain possible PRIs to meet the requirements of our real-time ESM system.

Suppose, the scout's scanning interval is  $T_{\max}$  and there are  $M$  simultaneously active radios with PRIs as  $\{PRI_k\}$ ,  $k=1, 2, \dots, M$ . When no pulse is missing,  $N_k = \lfloor T_{\max} / PRI_k \rfloor$  pulses from the  $k$ th radio are intercepted and its  $N_k$  pulses segment the period  $T_{\max}$  into  $N_k$  sections to form a comb whose tooth pitch is  $PRI_k$ . All the same length combs with different pitches and "random initial phase". And apparently, the smaller the PRI is, the denser the comb is, i.e., the less other pulses found in its comb slot. Therefore, by counting pulses that falls into a certain PRI comb slot, we can found peaks that correspond to low PRIs, and averagely, the corresponding peak appears firstly on  $c$  level SDIF, where

$$c = \left\lfloor \sum_k \frac{PRI_{\min}}{PRI_k} \right\rfloor \quad (6)$$

\* For example, the built-in functions provided by MATLAB© such as comparison, maximizing and minimizing and searching (find) -which are most frequently used in deinterleaving algorithms - would "expense" no single flop. That's why in newly published MATALB 6.0 and later obsoletes flop function.

† However no literature covers it.

where  $\lfloor \cdot \rfloor$  returns the floor of " $\cdot$ ". For example, given PRIs as in Fig.3,  $c = \lfloor 1 \ 2 \ 5 \rfloor$  thus 12000us appears in the 1<sup>st</sup> SDIF. And in case of Fig.4-a, it can also be predicted that the 1<sup>st</sup> SDIF tells 17331us.

Thus, if there are  $N$  radios of the same PRI and when no pulse missing occurs, the real PRI would only appear at the  $N$ th level SDIF.

In a word, in any case above we can always find peak in an SDIF of certain level, which equals to a real PRI. Our simulation and experiments show that the decision level is often lower than when use threshold-function algorithms [4] even with proper ( $x, k$ ) pairs. Moreover, the technique introduced involves only comparative and additive operations, thus lots of processor time can be saved.

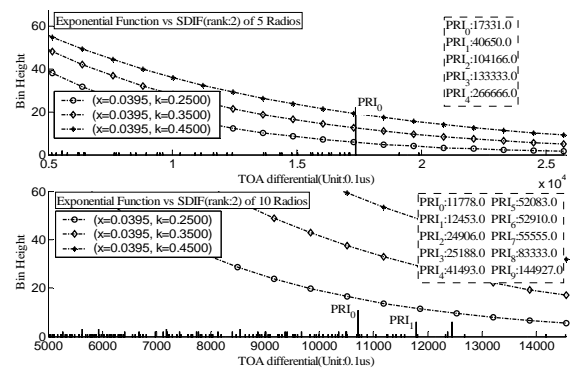


Fig.4 PRI estimation Performance of that in [4] in different Scenarios.

### 3.3. Deinterleaving Algorithm

As stated in Section 1, the proposed algorithm in this paper uses two pulse parameters: AOA and TOA. The deinterleaver begins with AOA-clustering, which breaks a huge pulse sequence into several smaller pulse sequences according to their different azimuths. Then for each azimuth group TOA-deinterleaving is performed. See Fig.5-(a). AOA-clustering is very important to the whole algorithm, for, similar to bi-search algorithm, clustering pulses from the same direction into smaller groups greatly reduces the total processing time of the TOA-deinterleaving algorithm; however, it is out of scope of this paper and only TOA clustering, or TOA-deinterleaving, is discussed.

The TOA-clustering algorithm needs two key parameters, which are highly dependent on the measure system and the radio environment. The first one is TOA precision (given by R.M.S error or maximum error) as window size used by SDIF forming and sequence searching process. The other one is pulse-missing ratio, which is given by

$$P_m = N_r / N_R \quad (7)$$

$$N_R = T \sum_k 1 / PRI_k$$

where  $N_R$  is the should-be received pulse number when no pulse missing occurs,  $N_r$  is the actual received pulse number,  $T$  is the sample interval, and

$PRI_k$  is PRI of the  $k$ th radio which can be estimated when deinterleaving completes.  $P_m$  is mainly determined by the ESM system features and used to withstand pulse missing in sequence searching process. However, margin had to be reserved for TOA measurement error, when PRI-jittered radars are active.

In the algorithm, the possible PRI is obtained when the SDIF calculation is completed, and the matched TOAs of a possible PRI are extracted when sequence searching is successfully completed. See Fig.5-(b, c). In this flow chart, we assume that 1) the max-pulse-missing ratio is 25%; 2) As a matter of fact, at least three pulses are required to form a regular PRI sequence; 3) Only one PRI estimated by peak-value-detection. Therefore, the deinterleaver deems that the estimated PRI is error when (a) we can not find the pulse matched the PRI successively for more five times, and (b) when the searching is over, if the number of hits is less than the expected pulse number determined by the PRI and the sample-time,  $T_{max}$ .

In Fig.5 **bW** is TOA difference clustering precision to form a histogram hit. After SDIF calculation, there would be **cLEN** clusters whose centers are in the vector **Cs**, and their bin heights in **Bs**. Moreover, the **MaxPosition** denotes the possible PRI with bin height of **MaxHeight**. Whenever a TOA difference falls into a cluster with bin height of **B**, the cluster center is replaced by the mean value of all the (**B**+1) TOA differences.

The sequence-searching algorithm shown in Fig.5-(c) is divided into two matching loops. The first two pulses (TOAs) are to be matched in the first **WHILE** loop, then, if successful, the pulses thereafter are to be traced in the second **WHILE** loop. In both loops, the integer variable **k** denotes the number of missing pulses between TOAs (j) and TOAs (i), if they both belong to the subsequence of the estimated **PRI**. When the next pulse is found matching with possible **PRI**, the **PRI** value is adjusted to withstand TOA error and estimation error.

## 4 Simulation Results

The algorithm has been implemented in both C and MATLAB language. Simulation in MATLAB language shows it is much better than threshold-function algorithms. Simulations in C language prove that the proposed algorithm successfully identifies emitters from a sample PDW sequence in its sample interval. The following tables give the average simulation results of 200 times when implemented in MATLAB 5.3 (Table 1) and in Microsoft Soft Visual C++ 6.0 (Table 2) on Intel PIII 550MHz CPU.

Furthermore, the algorithm has been implemented in a TMS320C5x-based ESM system.

Table 1 Performance in MATLAB

Radios	PRIs	PMR	Cost in flops	True PRI	False PRI
5	5	25%	8433	4	1
5	1	20%	3000	1	0
10	10	10%	15041	10	0
10	1	5%	11616	1	0

Sample Time=50ms

Table 2 Performance in C

Radios	PRIs	Sample Time	Cost in ms	True PRI	False PRI
5	5	100ms	1.15	5	0
5	1	100ms	3.20	5	0
10	10	100ms	5.80	10	0
10	1	100ms	17.20	1	0
15	5	100ms	22.85	5	0

PMR=5%

From Table 1, in the worst case in our approach spends 15041-flop, which is only 3 times lower than that of the best case in [4], and only 126 times lower than that of its worst case in [4] when only TOA used.

From Table 2, even when  $P_m$  is 5%, no error found. Moreover, the processor time used by the proposed algorithm in the worst case is only fifth the sample time, which means that during the each sample duration, much time is saved for further analysis on individual radios.

## 5 Conclusion and Further Discussion

In this paper, we analyzed performance especially efficiency of early proposed deinterleaving algorithms by D.J. Milojevic [4], and proposed the simple and straightforward SDIF-based PRI estimation scheme. Simulation results have shown that the approach performs much better than that given by [4] (in MATLAB implementation), and the C Language implementation also demonstrates itself competent for real-time processing in ESM systems.

Furthermore, the computation cost can be further minimized,

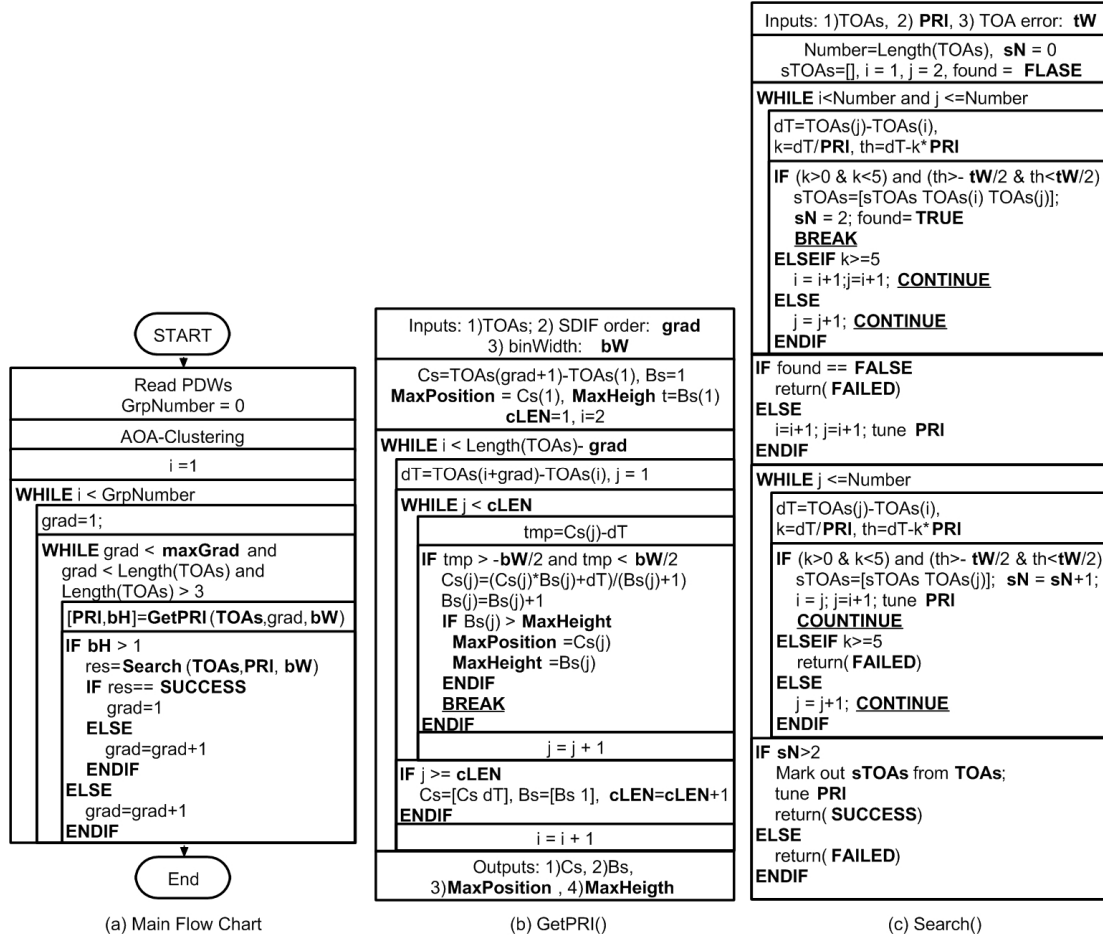
- 1) if we use not only PRIs corresponding to the peak in SDIF but also those corresponding to the secondary and tertiary peaks in SDIF;
- 2) if the smallest possible PRI value is already known, we can ignore TOA differences smaller than the smallest PRI in SDIF calculation.

The approach uses only simple operations like addition and comparison to calculate the SDIF and by peak-value-detection (PVD) to obtain possible PRI, thus no time-consuming threshold functions are involved and much time is saved – due to

### References:

- [1]. C.L. Davies, Ph.D., and P. Hollands, B.A. "Automatic processing for ESM" IEE Proc., Vol.129, Pt. F, No.3, June 1982, pp.164-171
- [2]. J.A.V. Rogers, Ph.D. "ESM processor system for high pulse density radar environments" IEE PROC., Vol.132, Pt. F, No.7, DECEMBER 1985, pp.621-624
- [3]. H.K. Mardia, BSc. PhD "New techniques for the deinterleaving of repetitive sequences" IEE PROCEEDINGS, Vol.136, Pt. F, No.4, AUGUST 1989

- [4]. D.J. Milojevic and B.M. Popovic “Improved algorithm for the deinterleaving of radar pulses” IEE PROC., Vol.139, Pt. F, No.1, FEBRUARY 1992 pp.98-104
- [5]. Jane Perkins and Ian Coat “Pulse Train Deinterleaving via the Hough Transform” IEEE Proc. ICASSP-92
- [6]. S.R. Deans, “Hough transform from the radon transform” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 3, March 1981 pp.195-188
- [7]. Hsuen-Chyun SHYU, Chih-Chi CHANG, Yueh-Jyun LEE and Ching-Hai LEE “Radar Signal Clustering and Deinterleaving by a Neural Network” IEICE Trans. Fundamentals Vol. E80-A, No. 5, May 1997 pp.903-911
- [8]. G. Noone, Radar pulse train parameter estimation and tracking using neural networks. In Proc. Inter. Conf. Artificial Neural Networks and Expert Systems, New Zealand, Nov. 1995, pp. 95-98.



(a) Main Flow Chart (b) GetPRI() (c) Search()

Fig.5 PRI estimation and TOA-deinterleaving based upon PVD in the SDIF