# Online Pulse Deinterleaving With Finite Automata

**ZHANG-MENG LIU**
National University of Defense Technology, Changsha, China

**Most of the existing deinterleaving methods work offline; they deinterleave pulse streams in a recursive way via multipass searching. Such methods do not fit online processing applications, and they have very low deinterleaving efficiency. In this paper, I address the deinterleaving problem of streams with repetitive periods, such as streams with constant or stagger pulse repetition intervals. Transitions between the pulses of different states are illustrated with regular grammars, and finite automata are established accordingly to realize *online* pulse deinterleaving. The states of pulse streams being deinterleaved are stored and continuously updated in the automata, and newly received pulses are judged by the associated finite state controls to determine whether they come from a certain emitter, so as to realize online deinterleaving. In the automaton-based method, multiple automata can be started to work parallel on the same stream, and they deinterleave intersected streams via one-pass (instead of recursive) searching. Simulation results also demonstrate the superiority of the proposed method in pulse-deinterleaving performances.**

Manuscript received August 24, 2018; revised May 7, 2019; released for publication June 18, 2019. Date of publication June 27, 2019; date of current version April 10, 2020.

DOI. No. 10.1109/TAES.2019.2925447

Refereeing of this contribution was handled by K. D. Pham.

This work was supported in part by the National Science Foundation of China under Grant 61771477 and in part by the Program for Innovative Research Groups of the Human Provincial National Science Foundation of China.

Author's address: Z.-M. Liu is with the State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, National University of Defense Technology, Changsha 410073, China, E-mail: (liuzhangmeng@nudt.edu.cn).



## I. INTRODUCTION

Pulse deinterleaving plays an important role in electronic intelligence and electronic support measurement systems [1]. Pulses with distinguishable frequencies and widths can be separated directly via clustering and categorization [2]. However, due to fast developments and widespread usage of various advanced communication [3], navigation [4], and radar [5] systems, the electromagnetic environment is becoming more and more crowded. Pulses of different radars are probably intersected and should be deinterleaved beforehand to facilitate subsequent processes, such as emitter recognition and localization. Ata'a and Abdullah [6] established a neural network to realize online clustering of different emitters' pulses according to their direction-of-arrival and radio frequency. Interpulse repetitions within the clusters are, then, analyzed to deinterleave the pulses of different radars [6]. This paper concentrates on scenarios in which intersected pulse streams coming from different emitters have unavailable or indistinguishable directions, frequencies, and widths, and it proposes to realize online deinterleaving according to pulse repetitive intervals (PRIs) [7]–[9].

PRI-based methods have been developed to deinterleave pulse streams with repetitive temporal structures. Among them, the two most widely cited methods are the cumulant difference histogram (CDIF) [10] and the sequential difference histogram (SDIF) [11]. They first analyze the distribution of differential time-of-arrival (DTOA) between pulses to estimate the repetitive period of the stream (named frame PRI), and they then use this period to deinterleave substreams with constant PRI. Multiple constant-PRI substreams can be intersected to recover streams of emitters with stagger PRIs.

The frame-PRI-based methods have two significant shortcomings. One is that they do not fit online deinterleaving tasks. That is because the estimation of frame periods can be realized only after collecting sufficient pulses, and the subsequent deinterleaving process based on the frame PRI estimates will be further delayed [10], [11]. In some long-time continuous deinterleaving applications, pulse streams should be divided into shorter periods to save processing and storage burdens. Frame PRIs are estimated in the first period, and the estimates can be exploited in the subsequent periods to realize sequential deinterleaving. This is a straightforward extension of the existing CDIF [10] and SDIF [11] methods. It addresses the online deinterleaving problem just to a very limited extent, as only one substream with constant frame-PRI can be deinterleaved online, while the remaining pulses of stagger streams and the streams of other emitters can only be deinterleaved afterward in an offline manner.

The second shortcoming of the batch deinterleaving methods is that they have very low processing efficiency. For streams consisting of substreams coming from different emitters, constant frame-PRI can be exploited to separate only one substream at a time. The separated substream is, then, extracted from the original stream, and the remaining stream is exploited to start a new deinterleaving process. Such substream deinterleaving processes can only be executed recursively via multipass searching. If the stream contains pulses of a large number of emitters, or if the stagger PRIs have high dimensions, the stream should be searched many times before being deinterleaved completely, which further deviates online processing requirements.

Some other theoretically intense deinterleaving methods have also been proposed, e.g., Kalman filter based methods [12], [13], hidden Markov model based methods [14], [15] and multiple hypothesis tracking based methods [16]. They introduce more in-depth academic research works into the deinterleaving process, but none of them is designed for the online deinterleaving problem. Moreover, more complicated mathematical processing in the algorithms further lowers down the computational efficiency of deinterleaving.

In this paper, I address the online pulse-deinterleaving problem based on the prior information of emitters' PRI patterns, which may be available in the knowledge databases of electronic support systems or have been extracted previously during long-time continuous deinterleaving. The proposed method does not follow the guidelines of statistical frame-PRI analyzing and recursive deinterleaving, but it treats the problem of deinterleaving sequential pulses as one of symbol-string recognition and concentrates on meeting the requirement of *online* deinterleaving. The string recognition induced deinterleaving idea makes use of not only information about frame PRI but also detailed transition rules between different pulses. The transition rules are exploited to determine whether the newly received pulses match with the previously deinterleaved substreams. The matching process is implemented automatically using automata in a way similar to that of sequential string recognition. In order to explain the feasibility of this idea, I first illustrate the regular grammar characteristics of streams having repetitive periods, and I, then, establish finite automata accordingly to realize pulse deinterleaving. Multiple automata are, then, assembled in parallel to deinterleave intersected streams online via one-pass searching.

The rest of this paper consists of six parts. Section II introduces the problem of pulse deinterleaving. Section III illustrates a regular grammar for pulse streams with repetitive structures and establishes automata for pulse deinterleaving. Section IV proposes an automaton-based online pulse-deinterleaving method for a single emitter. Section V extends the method to realize parallel online deinterleaving of pulses from multiple emitters. Section VI carries out simulations to demonstrate the performance of the proposed method. Section VII concludes this paper.

## II. PULSE STREAM FORMULATION

Pulse streams are not merely the clusters of independent pulses. Interpulse relations, especially PRIs, introduce abundant information into streams. PRI is the main parameter for deinterleaving the intersected pulses of different emitters when their frequencies and pulsewidths are indistinguishable [1], [10], [11]. Streams with constant or stagger PRIs have repetitive periods, which is named frame PRI in this paper. Pulses within a frame have different contexts, and they depart from preceding and successive pulses differently. These pulses are annotated with different states. Suppose a repetitive pulse stream has $M$ states, namely $q_1, q_2, \ldots, q_M$, with $M = 1$ for constant PRI streams and $M > 1$ for stagger PRI streams. The PRI between a pulse with state $q_i$ and its successive pulse is denoted by $PRI_i\,(i = 1, 2, \ldots, M)$, as shown in Fig. 1. The state transitions in the pulse stream is depicted in the following:

$$q_1 \xrightarrow{PRI_1} q_2 \xrightarrow{PRI_2} \cdots \xrightarrow{PRI_{M-1}} q_M \xrightarrow{PRI_M} q_1 \xrightarrow{PRI_1} \cdots . \tag{1}$$

Transitions between the pulses of different states are periodic. The DTOAs between adjacent pulses have typical values of $PRI_1, \ldots, PRI_M$, and higher order DTOAs exist between farther apart pulses, which are the summations of successive PRIs, such as $PRI_1 + PRI_2$ and $PRI_2 + PRI_3 + PRI_4$. For streams with repetitive periods, their $M$th order DTOA has a particular value, which is called the frame PRI of the stream, i.e.,

$$T_0 = \sum_{i=1}^{M} PRI_i. \tag{2}$$

This DTOA exists between the pulses with the same states but located in adjacent frames. It emerges most frequently among all the DTOAs in noise-free repetitive streams. The DTOAs between the pulses with the same states and located in farther apart frames are the harmonics of the frame PRI, i.e., $2T_0,\ 3T_0, \ldots$. These harmonic frame PRIs also emerge frequently in higher order DTOA clusters. They cause great difficulties in the analysis and recognition of the frame PRI [11].

Streams collected in practical systems are usually contaminated by various noises, such as missing pulses and intersected streams. These noises disturb the state transition sequence in Fig. 1, and the typical PRI values of $PRI_1, \ldots, PRI_M$ and the transition pattern between them become indistinct. However, interpulse DTOA values of the same emitter remains unchanged in noisy streams, with $T_0$ and its harmonics being highlighted statistically in DTOA histograms [10], [11]. A straightforward way for analyzing the frame PRI is differentiating the pulses' time-of-arrival (TOA) and clustering the DTOAs along the time axis with a histogram. A great number of DTOAs will cluster around $T_0$ and its harmonics, which provides the information for recognizing and estimating the frame PRI. The frame PRI estimate can, then, be exploited to extract the substreams with constant PRI of $T_0$ from the original stream. The substreams with the same frame PRI are, then, joined together to recover stagger streams. This is how CDIF [10] and SDIF [11] realize pulse deinterleaving.

The above-mentioned deinterleaving methods are based on statistical repetitive period analysis, and they adapt to offline tasks only. That is because frame PRIs can be estimated stably only when a large number of pulses have

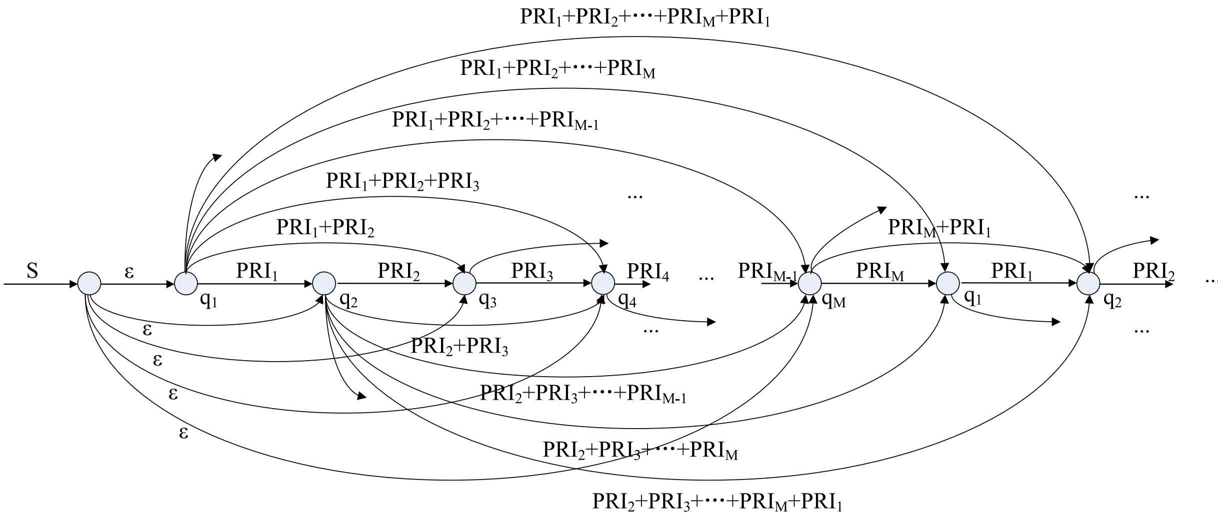


Fig. 1. Pulse state transitions of streams with repetitive periods.

been collected, and substreams should be extracted recursively afterward based on the frame PRI estimates. When deinterleaving an intersected pulse stream consisting of the substreams of $K$ emitters, with each substream having $M$ states, the stream should be searched for $K \times M$ times to deinterleave the substreams of all the emitters. The time delay of the recursive deinterleaving process will be very large, making these methods unable to fit online deinterleaving tasks.

## III. AUTOMATA ESTABLISHING FOR PULSE STREAM RECOGNITION

If prior information about the PRI patterns is available for an emitter, the deinterleaving problem can be treated as pulse train recognition, one along the time axis, which is similar to symbol-string recognition according to a certain grammar. In the latter problem, grammar regulations are illustrated and exploited to determine whether new symbols make up a correct string [17]. When the string ends and the state reaches a valid terminal, the string is judged as a correct one. Similarly, by describing the pulse states and their transition patterns in a regular formulation, the newly arriving pulses can also be judged to see whether they belong to a certain stream. In this way, correct pulses are reserved and noises are excluded instantaneously at the time when they arrive, and the deinterleaving process can be implemented online and completed immediately once the stream ends.

A useful tool for string recognition is automaton [17]. In this section, I first illustrate the streams having repetitive structures with regular grammars, and I, then, introduce the principles of common automata to design special automata for pulse stream recognition. These automata are finally used to solve *online* pulse-deinterleaving problems.

### A. Regular Grammars of Repetitive Pulse Streams

In Fig. 1, the pulse state transition sketch mainly contains the information about how states transfer between each other, and also contains the DTOA for different transitions. In this paper, pulse deinterleaving is realized mainly by exploiting the DTOA information, and other pulse features, such as frequencies and widths, are embedded in the states for transition validation. The interpulse state transition regulations can be described with the following grammar [17]:

$$G = (V, T, P, S) \tag{3}$$

where $V = \{q_1, q_2, \ldots, q_M\}$ stands for the pulse state set that corresponds to the variable set in common string grammars; $T = \{\mathrm{DTOA}(q_i, q_j), 1 \le i, j \le M\}$ contains all the possible DTOAs of the pulse stream when transferring between different states, which correspond to the terminals in common grammars; $S = V$ represents the starting state set. As streams may begin with any state in $V$, a blank-move of $\epsilon$ is added in Fig. 1, so as to transfer the starting state to any one of $q_1, q_2, \ldots, q_M$. Furthermore, $P = \{q_j \to [\mathrm{DTOA}(q_i, q_j), q_i], 1 \le i, j \le M, \text{or } q_j \in S\}$ is the production equation set, which indicates temporal transferring relations between temporally adjacent pulses. Each production equation means that by backing off a pulse stream in state $q_j$ with a DTOA of $\mathrm{DTOA}(q_i, q_j)$ along the time axis, the state of the stream retrogresses to $q_i$. In noise-free streams, the indexes of the two states in each production equation satisfy $j = i + 1\,(1 \le i < M)$ or $i = M$, $j = 1$, and $\mathrm{DTOA}(q_i, q_j) = PRI_i$. Practical streams are contaminated by noises such as missing pulses; thus, $i$ and $j$ may take any value in $\{1, 2, \ldots, M\}$, and $\mathrm{DTOA}(q_i, q_j)$ may be larger than the maximal value in $PRI_1, \ldots, PRI_M$, or even larger than the frame PRI of $T_0$.

The grammar in (3) contains only a finite number of states, and each production equation has a structure of $A \to \omega$ or $A \to \omega B$ with $A, B \in V$ and $\omega \in T$. It can be concluded from (3) that the states of the pulse streams with repetitive structures transfer in a similar way as that in regular grammars [18], [19], [20]. As each string generated by regular grammars can be recognized using a certain finite automaton (FA) [21], I propose to recognize sequential

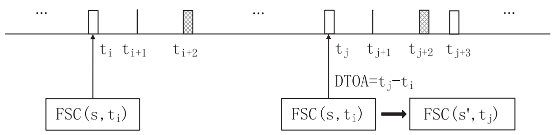


Fig. 2. Physical model of pulse-deinterleaving FA. Pulses having different shapes and fillings come from different emitters.

pulses from the same emitter with specially designed automata.

### B. Finite Automata for Repetitive Pulse Stream Recognition

For the regular grammar $G$ of the pulse streams in (3), the corresponding FA [17], [21] is as follows:

$$Z = (Q, \Sigma, \delta, q_0, F) \tag{4}$$

where $Q = V = \{q_1, q_2, \ldots, q_M\}$ contains all possible pulse states; $\Sigma = T = \{\mathrm{DTOA}(q_i, q_j), 1 \le i, j \le M\}$ represents the DTOAs of adjacent pulses in the presence of data noises, which corresponds to the input alphabet of automata; $\delta$ is the state transition function describing how an automaton transfers from one state to another after receiving a pulse with particular parameters. The transition function corresponds to the production equation set in $G$, which defines interpulse DTOAs from state $q_i$ to state $q_j$. Furthermore, $q_0 \in Q$ denotes the initial state set of the automaton. $F = \$$ stands for the final state of the automaton and also the termination of the stream.

The physical model of an FA for pulse deinterleaving is shown in Fig. 2. Two basic components are contained in the automaton, one is an input pulse stream, and the other is a finite state control (FSC), which stores the current automaton state and interstate transition regulations. The pulse stream is inputted into the FSC sequentially, and the FSC controls a read head to read in the pulses one by one. Once a pulse is read in and has been processed, the FSC shifts the read head to the next pulse. The FSC of the proposed pulse-deinterleaving FA has a slightly different structure from that of common string-recognition FAs. It stores not only the current state and state transition regulations but also the TOA of the preceding pulse. The modification is necessary because state transition regulations rely on interpulse DTOAs.

In Fig. 2, it is supposed that the FA has successfully deinterleaved a pulse at time $t_i$, and the FSC state is updated to $s$. The TOA of this pulse is stored in the FSC. After that, the read head of the FSC is shifted rightward. When a new pulse arrives, its frequency and width, together with its DTOA with respect to (w.r.t.) $t_i$, are read into the FSC for checking. The pulses arriving at time instants of $t_{i+1}$ and $t_{i+2}$ are not accepted because they do not belong to the same stream with the pulse at time $t_i$, and, thus, their DTOAs w.r.t. $t_i$ do not match the state transition function $\delta$ of the FA. The transition function is satisfied at time $t_j$ for the first time after time $t_i$, so the pulse is accepted by the FSC, and it is associated with the substream being deinterleaved. In addition, the FSC state is updated from $s$ to $s'$, and the stored TOA is also updated from $t_i$ to $t_j$, while the transition function $\delta$ remains unchanged.

As new pulses arrive one after another, the FA repeats the above-mentioned pulse-recognition and state-updating processes successively. The substream satisfying the transition function will be extracted, and the deinterleaving task is completed online once the pulse stream ends. The state transition function provides important criteria for judging whether new pulses come from the same emitter.

The state transition function corresponding to the pulse stream in Fig. 1 can be described more clearly as shown in Table I. In the table, $s$ and $s'$ represent two adjacent FSC states, and the table contents stand for interpulse DTOAs when transferring between different states. Suppose that the current FSC state is $q_i$, and a new pulse arrives after time $\mathrm{DTOA}(q_i, q_j)$, which is the DTOA on the $q_i$th row and the $q_j$th column in Table I, with its frequency and width matching the parameters of state $q_j$, then the FSC transfers from state $q_i$ to state $q_j$ and the stored time is updated to $t_j$.

In Table I, only the DTOAs that are smaller than the frame PRI are listed. Actually, in streams when a great number of pulses are missed, the DTOAs may be larger than the frame PRI, and the interpulse DTOA should satisfy the following formulation in general:

$$\begin{aligned}&\mathrm{mod}\,(\mathrm{DTOA}(q_i, q_j), T_0)\\&\quad = \mathrm{mod}\left(\sum_{l=i}^{M+j-1} PRI_{\mathrm{mod}\,(l-1,M)+1}, T_0\right).\end{aligned} \tag{5}$$

That is, the DTOA between states $q_i$ and $q_j$ equals the summation of all the PRIs associated with the missed states between them, with $q_i$ included and $q_j$ excluded. In the equation, $\mathrm{mod}(a, b)$ represents the remainder of $a$ when divided by $b$. The upper index of the summation on the right-hand side of (5) is modified from $j-1$ to $M+j-1$, so as to include the cases in which $j \le i$; i.e., the FSC transfers from a latter state in a preceding frame to an earlier state in a subsequent frame. The cases in which interpulse DTOAs are larger than the frame PRI are covered by the remainder operator $\mathrm{mod}(\bullet, T_0)$. In practice, matching tolerance should be included to relax the rigorous equality in (5) due to TOA measuring noises, which will be taken into consideration in the simulations.

## IV. ONLINE DEINTERLEAVING OF PULSE STREAMS WITH FINITE AUTOMATA

When using the FA to deinterleave noisy pulse streams, two processes of FA start-up and FA updating are included, as shown in Fig. 3.

### A. Start-up of Finite Automata

During online pulse-deinterleaving processes, if a new pulse stream emerges at some time, its pulse TOAs will not match the state transition functions of other emitters' automata that have already been started; hence, the pulses will be skipped over by these automata. When the number of retained pulses exceeds a certain threshold, the most

TABLE I
Interpulse DTOAs in the State Transition Function

| s \ DTOA s' | $q_1$ | $q_2$ | $q_3$ | $\cdots$ | $q_M$ |
|---|---|---|---|---|---|
| $q_1$ | 0 | $PRI_1$ | $PRI_1 + PRI_2$ | $\cdots$ | $PRI_1 + \cdots + PRI_{M-1}$ |
| $q_2$ | $PRI_2 + \cdots + PRI_M$ | 0 | $PRI_2$ | $\cdots$ | $PRI_2 + \cdots + PRI_{M-1}$ |
| $q_3$ | $PRI_3 + \cdots + PRI_M$ | $PRI_3 + \cdots + PRI_M + PRI_1$ | 0 | $\cdots$ | $PRI_3 + \cdots + PRI_{M-1}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $q_M$ | $PRI_M$ | $PRI_M + PRI_1$ | $PRI_M + PRI_1 + PRI_2$ | $\cdots$ | 0 |

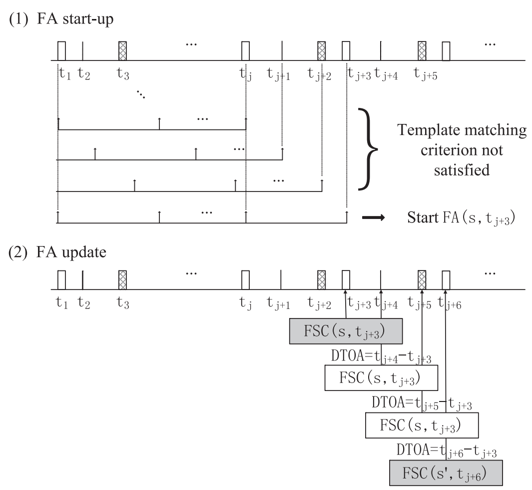


Fig. 3. Pulse-deinterleaving FA contain two processes: FA start-up and FA update.

recent pulses within a temporal window $\tau$ will be picked out to determine whether a new automaton should be started or not. The last pulse in the window is taken as a baseline, and the DTOAs of the preceding $J-1$ pulses within the temporal window w.r.t. it are computed in a reverse order along the time axis. The DTOA sequence is denoted by $0, -DTOA_1, \ldots, -DTOA_{J-2}, -DTOA_{J-1}$. If a new emitter's pulse stream is contained in the temporal window, this DTOA sequence will be highly correlated with another DTOA sequence, which is obtained from the emitters' state transition sketch, as shown in Fig. 1, by computing interpulse DTOAs similarly w.r.t. a selected baseline pulse.

As the pulse stream has $M$ different states, $M$ DTOA sequences can be obtained by choosing different baseline pulses. Each DTOA sequence is defined as a template (TP), and the $M$ TPs w.r.t. a baseline pulse with states from $q_1$ to $q_M$ are listed as follows:

$$
\begin{aligned}
&q_1 : 0, -PRI_M, -(PRI_M + PRI_{M-1}), \ldots \\
&q_2 : 0, -PRI_1, -(PRI_1 + PRI_M), \ldots \\
&\vdots \\
&q_M : 0, -PRI_{M-1}, -(PRI_{M-1} + PRI_{M-2}), \ldots.
\end{aligned} \tag{6}
$$

All the above sequences should be cut off within a temporal window of length $\tau$.

If the measured DTOA sequence matches well with the $m$th TP of a certain emitter, this emitter's pulse stream is believed to exist in the retained pulse sequence. A new FA corresponding to this emitter can then be started, and its starting state is set to $q_m$. The matching threshold depends on the stream PRI and the pulse missing probability. In the simulations in Section VI, the threshold is set to $\lfloor 0.5 \times \frac{\tau}{T_0/M} \times (1-\rho_{\text{miss}}) \rfloor$, where $\rho_{\text{miss}}$ represents pulse missing probability and operator $\lfloor a \rfloor$ outputs the largest integer not larger than $a$.

As is shown in Fig. 3(1), after the retained pulse number has exceeded, a predefined threshold, the TP matching window, is renewed according to "latest in, earliest out" criterion every time a new pulse arrives. The reverse DTOA sequence is, then, updated accordingly, and the updated DTOA sequence is matched with each candidate emitter's DTOA TPs. In Fig. 3(1), the TP set that is used for matching contains all the TPs of all candidate emitters, and the best matching result is used for starting an FA when necessary. At all pulses' arriving time instants before $t_{j+3}$, the number of matching pulses is smaller than a preset threshold; thus, no automata is started. When a new pulse arrives at time $t_{j+3}$, the matching pulse number between the measured pulse train and the $m$th TP equals the threshold. When this happens, the matched pulse sequence is extracted to form a new substream, and the control state of the newly started automaton is set to FSC($q_m$, $t_{j+3}$).

## B. Sequential Updating of Finite Automata

Once an automaton has been started, the state of the most recent pulse and its arriving time, i.e., $s$ and $t$, are recorded in the control of the automaton. When a new pulse arrives, the FSC evaluates the pulse's parameters and its DTOA w.r.t. the stored time instant, so as to judge whether this pulse fits the state transition function given in (4). If the transition criteria are satisfied, this pulse is accepted and added to the deinterleaved substream, and the FSC's states are updated. Otherwise, this pulse is skipped over, and the FSC's states are kept unchanged. After making the decisions and carrying out the actions, the read head of the FSC is shifted to the next pulse.

As is shown in Fig. 3(2), an automaton is started after receiving the pulse at time instant $t_{j+3}$, and the starting state is $s$. After that, two pulses arrive successively at time instants $t_{j+4}$ and $t_{j+5}$; their DTOAs w.r.t. $t_{j+3}$, i.e., $t_{j+4} - t_{j+3}$ and $t_{j+5} - t_{j+3}$, do not satisfy the interpulse DTOA criterion in (5), so they are skipped over by the automaton and retained for future processing. When another new pulse arrives at time $t_{j+6}$, its DTOA is $t_{j+6} - t_{j+3}$, which satisfies

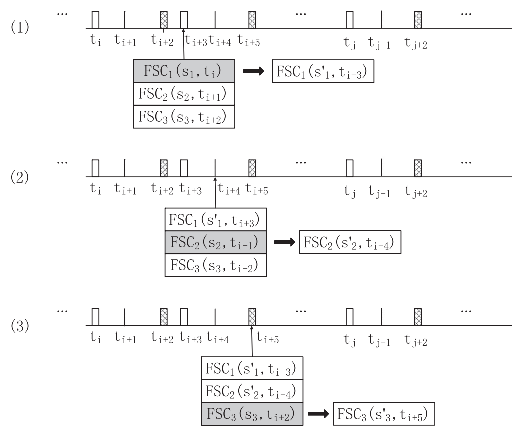


Fig. 4. Parallel deinterleaving with multiple automata.

(5); thus, this pulse is accepted and added to the associated substream. The automaton's state is, then, updated to $s'$ according to the new pulse, and the most recent pulse time is substituted by $t_{j+6}$.

## V. PARALLEL DEINTERLEAVING OF INTERSECTED PULSE STREAMS

It has been shown in the previous section that automata are able to realize *online* pulse deinterleaving for a single emitter. But, this property is not exclusive for the automaton-based deinterleaving method. Based on the prior information about the frame PRI, the deinterleaving methods in CDIF [10] and SDIF [11] can also deinterleave a single emitter's pulse stream with constant PRI online. However, multiple automata can be joined together to work in parallel, so as to deinterleave *online* intersected streams from multiple emitters with more complicated PRI structures, such as stagger PRIs. This is a more significant and exclusive superiority of the automaton-based method when compared with the existing counterparts.

Deinterleaving automata store substream states and the most recent pulses' time instants to judge whether newly arriving pulses belong to the associated substreams. If multiple automata are started on the same pulse stream and if each automaton has its own state, each of them just extracts particular pulses that obey their own state transition regulations, so multiple substreams radiated by different emitters do not interfere with each other. These automata act like pulse classifiers. The newly received pulses are classified and assigned to the right substream once they arrive. As is shown in Fig. 4, when a new pulse arrives, the automaton set integrates all recognition results of each automaton, and it makes a decision to associate this pulse with a certain substream and updates the corresponding automaton's states, or leaves it for future processing.

In Fig. 4, three automata are supposed to have been started before time instant $t_{i+3}$, and their FSC states at that time instant are $\mathrm{FSC}_1(s_1, t_i)$, $\mathrm{FSC}_2(s_2, t_{i+1})$, and $\mathrm{FSC}_3(s_3, t_{i+2})$, respectively. When a new pulse arrives at time $t_{i+3}$, it is processed by the three automata independently according to its parameters and its DTOA w.r.t. the

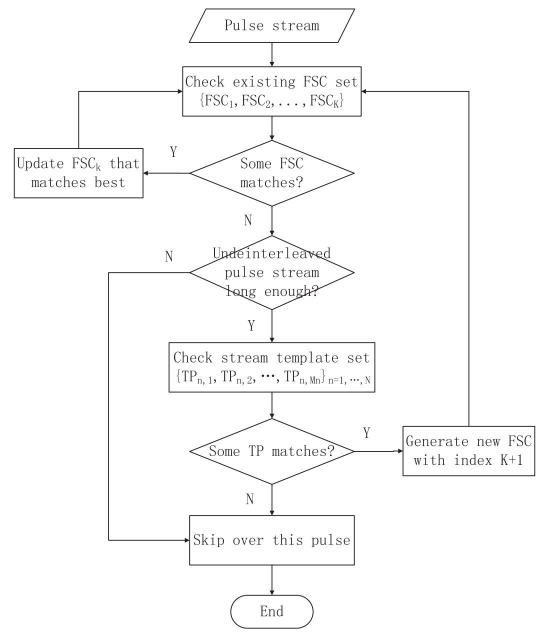


Fig. 5. Flowchart of parallel deinterleaving with multiple automata.

stored time instants in each automaton. This pulse matches best with the first automaton, and the matching degree satisfies the state transition criteria of the corresponding FSC, so this pulse is added to the associated substream, and the automaton's state is updated to $\mathrm{FSC}_1(s_1', t_{i+3})$. Similarly, the pulses arriving at time $t_{i+4}$ and $t_{i+5}$ are accepted by the second and third automata, respectively, and their FSC states are updated to $\mathrm{FSC}_2(s_2', t_{i+4})$ and $\mathrm{FSC}_3(s_3', t_{i+5})$ afterward.

Following the parallel deinterleaving guideline, the intersected pulse stream can be separated into multiple substreams via one-pass searching. Each of the separated substream contains pulses with their features and interpulse DTOAs obeying the regulations defined by the corresponding automaton. When comparing this method with the existing frame-PRI-based recursive deinterleaving methods [10], [11], the former is an online one being able to deinterleave successive pulse trains instantaneously, while the latter can only be implemented offline. The proposed method also has much higher processing efficiency. It searches the pulse stream only once, instead of $\sum_{k=1}^{M} M_k$ times for the frame-PRI-based methods, with $K$ being the emitter number and $M_k$ the state number of the $k$th emitter.

Similar to the single-automaton-based deinterleaving method shown in Fig. 3, each of the automata in the parallel deinterleaving method in Fig. 4 also consists of a start-up process and a continuous updating process. Both the processes of every automaton work in the same way as that shown in Figs. 3(1) and 3(2), respectively, so their details are not included here. By synthesizing the starting-up processes of each automaton into the parallel deinterleaving sketch in Fig. 4, the complete parallel deinterleaving flowchart can be obtained as that shown in Fig. 5.

In the flowchart, $K$ automata are supposed to have been started with an FSC set $\{\mathrm{FSC}_1, \mathrm{FSC}_2, \ldots, \mathrm{FSC}_K\}$. When a new pulse arrives, each of the automata evaluates the pulse according to its TOA, frequency, width, etc. independently. If the pulse's parameters satisfy some of the FSC's state transition criteria, the state of the FSC that matches best is updated, and the read head of the combined FA is shifted to the next pulse. Otherwise, if no state transition criterion is satisfied, the pulse is put into the cache. The retained pulse sequence is checked whenever a new pulse is added in, and if the temporal span of the cached pulses is shorter than a preset threshold, the read head moves to the next pulse directly without making any actions. Otherwise, if the temporal span is long enough, the newly received pulse is taken as a baseline to compute the DTOA sequence reversely along the time axis within a window of $\tau$. The DTOA sequence is compared with all $N$ candidate emitters' DTOA TPs, as shown in (6). The TP set is denoted by $\{TP_{n,1}, TP_{n,2}, \ldots, TP_{n,M_n}\}_{n=1,N}$ with $M_n$ being the state number of the $n$th emitter. If one TP matches best and if the matching pulse number exceeds a predefined threshold, a new automaton with an FSC of $\mathrm{FSC}_{K+1}$ will be started, and its initial state is set accordingly. The new automaton is included to expand the automata set to $\{\mathrm{FSC}_1, \mathrm{FSC}_2, \ldots, \mathrm{FSC}_{K+1}\}$. Otherwise, if no DTOA TP matches well enough with the measured DTOA sequence, the current pulse is cached, and the next pulse will be read in.

## VI. SIMULATION RESULTS AND ANALYSES

In this section, simulations are carried out to demonstrate the deinterleaving performance of the proposed method. The pulse stream to be deinterleaved is an intersected one containing the substreams of multiple emitters. All the substreams have stagger PRIs, and some pulses of each substream are missed randomly according to probability $\rho_{\text{miss}}$. The DTOAs are exploited as a main feature to realize pulse deinterleaving, and the other features such as pulse frequencies and widths are not considered to facilitate description. However, joint exploitations of these features are the straightforward extensions of the proposed method. The TOA measurement of each pulse has random noises with a standard deviation of 1 $\mu$s. DTOA tolerance of 5 $\mu$s is included when using (5) for forward pulse association. The length of the temporal window used for matching measured DTOA sequences and TPs, i.e., $\tau$, is set to be 30 000 $\mu$s.

In the following, several simulations will be designed to show how different factors influence the performances of different deinterleaving methods. The deinterleaving performance is evaluated according to the total deinterleaving probabilities of all the intersected substreams. The frame-PRI-based deinterleaving method used by the CDIF [10] and the SDIF [11] after PRI estimation is listed for performance comparison, which is named *DIF-based* method. The proposed method is named *automaton-based* method. Each simulation result is obtained from 1000 Monte Carlo simulations.

In the first group of simulations, five emitters' substreams with the same stagger PRIs of [700 $\mu$s, 800 $\mu$s, 900 $\mu$s, 1000 $\mu$s] intersects together. The prior estimate of each stagger PRI has a random measurement error with mean 0 and a standard deviation of 0.2 $\mu$s. The pulse missing probability of $\rho_{\text{miss}}$ is supposed to increase from 0% to 80%. In each simulation, the measurement errors of each stagger PRI and each pulse's TOA are generated randomly; the initial states of the substreams and intersubstream time shifts are also set randomly. The deinterleaving probabilities of the two methods are shown in Fig. 6.

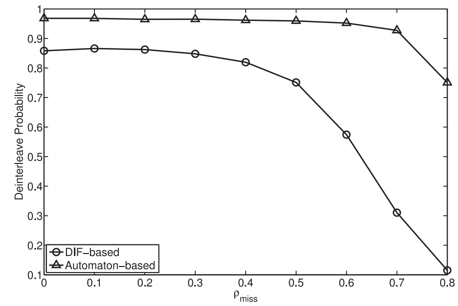


Fig. 6. Correct deinterleaving probabilities for different pulse missing probabilities.

It can be concluded from Fig. 6 that the proposed method has higher deinterleaving probabilities than those of its counterpart in all the cases considered, and as the pulse missing probability increases, its superiority further enhances. The major reason is that the two methods realize deinterleaving via forward pulse association. In the proposed method, temporally adjacent pulses are selected at each time, but the DIF-based method selects pulses that depart from the current one by a DTOA of one or more frame periods, which probably contain more than one adjacent DTOAs. Therefore, the stagger PRI measurement errors cumulate in the DIF-based method, and this method associates incorrect pulses more easily. As the pulse missing probability increases, the intervals of adjacent pulses contain more stagger PRI components, and the effect of the cumulated TOA measurement errors, thus, aggravates to deteriorate the performances of both the methods. But, as the DIF-based method always associates pulses that are farther away than the automaton-based method, the performance of the former deteriorates more significantly.

In the second group of simulations, $\rho_{\text{miss}}$ is fixed at 60%, and the intersected substream number increases from 2 to 8, while the other parameters are kept unchanged; then, the deinterleaving performances of the two methods are shown in Fig. 7. The deinterleaving probabilities of both the methods decrease slightly as the emitter number increases, and they have similar decreasing rates, which is caused majorly by the increase of pulse densities.

Then $\rho_{\text{miss}}$ is fixed at 60%, and the intersected substream number is 5, while the dimension of the stagger PRIs changes. The stagger PRI values increase from 700 to 1300 $\mu$s with an interval of 100 $\mu$s, and a certain number of the first values are selected to form stagger PRI patterns of different dimensions, e.g., second-dimensional stagger PRI pattern is [700 $\mu$s, 800 $\mu$s], fourth-dimensional

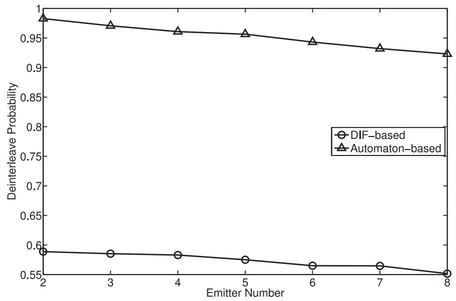


Fig. 7. Correct deinterleaving probabilities for different numbers of intersected substreams.

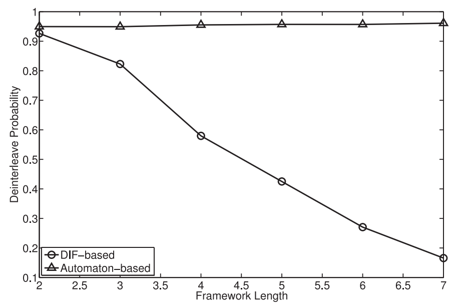


Fig. 8. Correct deinterleaving probabilities for different stagger dimensions.

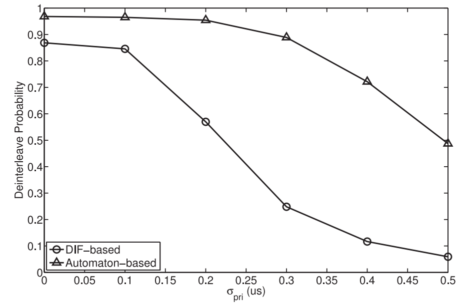


Fig. 9. Correct deinterleaving probabilities for different PRI measurement errors.

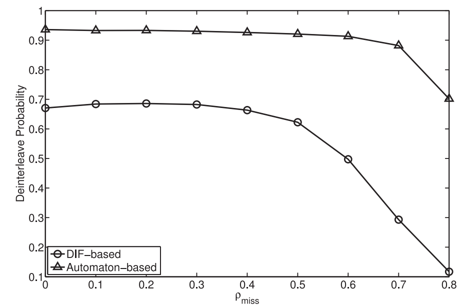


Fig. 10. Correct deinterleaving probabilities for different pulse missing probabilities when intersected substreams have different stagger PRI's.

stagger PRI pattern is [700 $\mu$s, 800 $\mu$s, 900 $\mu$s, 1000 $\mu$s], etc. The other simulation parameters are kept unchanged, and the simulation results are shown in Fig. 8.

The results indicate that the performance of the automaton-based method is rarely influenced by the stagger dimension. That is because the most recent pulse states are stored in the FSC during the deinterleaving process, and the performance of the deinterleaving depends only on the association precision of adjacent pulses, which is immune to the stagger dimension and insensitive to detailed stagger PRI values. However, the performance of the DIF-based method depends on the measurement precision of the frame PRI, which deteriorates with increasing stagger dimensions; thus, its deinterleaving probability decreases fast with the stagger dimension.

In the following group of simulations, I fix $\rho_{\text{miss}}$ at 60%, the intersected substream number at 5, the stagger dimension at 4, and increase the standard deviation of the PRI measurement error denoted by $\sigma_{\text{pri}}$ from 0 to 0.5 $\mu$s, while the other parameters are kept unchanged. The deinterleaving performances of the two methods are shown in Fig. 9. The results indicate that the deinterleaving probabilities rely heavily on this measurement error. That is because this error directly introduces biases between DTOA association regulations used in the deinterleaving methods and the practical DTOA measurements. The larger is the value of $\sigma_{\text{pri}}$ is, the more difficult is for the methods to realize precise pulse associations. As the DIF-based method exploits the frame PRI estimate, which accumulates more than one stagger PRI measurement errors, its deinterleaving regulation deviates more largely from practical DTOA measurements; thus its performance deteriorates more rapidly with $\sigma_{\text{pri}}$.

Finally, the simulation settings associated with Fig. 6 are modified to design another group of simulations. The emitter number is increased to six, which divides into two groups. The first 3 emitters' stagger PRIs are [700 $\mu$s, 800 $\mu$s, 900 $\mu$s, 1000 $\mu$s], and the second 3 emitters' stagger PRIs are [1000 $\mu$s, 900 $\mu$s, 800 $\mu$s, 700 $\mu$s]. The other simulation settings are kept unchanged, and the simulation results are shown in Fig. 10.

Comparing the probabilities in Figs. 6 and 10, one can conclude that the performance of the automaton-based method deteriorates only slightly, which is caused mainly by the effect of increasing emitter number. In contrast to its satisfying performance, the deinterleaving probability of the DIF-based method lowers down largely by about 20%. As a result, the proposed method surpasses the DIF-based one more significantly in Fig. 10 than that in Fig. 6. A major reason causing the differences is that the two groups of emitters have the same frame PRIs; thus, the DIF-based method may be confused in selecting the right pulses for certain intersubstream time-shifts. On the contrary, the automaton-based method makes use of not only the frame PRI but also the contexts of each stagger PRI. The contexts make the pulses of different substreams more distinguishable.

## VII. CONCLUSION

This paper addresses the *online* deinterleaving problem of streams with repetitive structures based on the prior information of PRI patterns. The state transition regulations of the streams are illustrated with regular grammars, and an automaton-based deinterleaving method is, then, proposed accordingly. The FSC of the automaton stores not only the interpulse state transition regulations of the pulse stream but also the stream state and the TOA of the most recent pulse. In this way, the automaton is able to judge whether a newly received pulse belongs to the substream associated with this automaton and, thus, solves the deinterleaving problem in an online manner. The superiority of the automaton-based method that should be highlighted is that multiple automata can be joined together to work in parallel and deinterleave an intersected pulse stream via one-pass searching. Its processing efficiency is, thus, significantly higher than that of the existing recursive deinterleaving methods, which separates substreams from the intersected stream one after another. The simulation results also show that the proposed method makes better use of the PRI contexts between adjacent pulses, and it is influenced less significantly by accumulating PRI measurement errors than previous frame-PRI-based methods. It surpasses its recursive deinterleaving counterparts largely in the cases of different pulse missing probabilities, different numbers of intersected substreams, different stagger dimensions, and different PRI measurement errors.

REFERENCES


[1] R. G. Wiley
*ELINT: The Interception and Analysis of Radar Signals*. Norwood, MA, USA: Artech House, 2006.

[2] J. Liu, J. P. Y. Lee, L. Li, Z.-Q. Luo, and K. M. Wong
Online clustering algorithms for radar emitter classification
*IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1185–1196, Aug. 2005.

[3] H. Arslan
*Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems*. New York, NY, USA: Springer-Verlag, 2007.

[4] D. C. Robbins, R. K. Sarin, E. J. Horvitz, and E. B. Cutrell
Advanced navigation techniques for portable devices
U.S. Patent 7 327 349, Feb. 5, 2008.

[5] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin
*Principles of Modern Radar*. Citeseer, 2010.

[6] A. W. Ata'a and S. N. Abdullah
Deinterleaving of radar signals and PRF identification algorithms
*IET Radar Sonar Navig.*, vol. 1, no. 5, pp. 340–347, 2007.

[7] P. S. Ray
A novel pulse TOA analysis technique for radar identification
*IEEE Trans. Aerosp. Electron. Syst.*, vol. 34, no. 3, pp. 716–721, Jul. 1998.

[8] R. J. Orsi, J. B. Moore, and R. E. Mahony
Spectrum estimation of interleaved pulse trains
*IEEE Trans. Signal Process.*, vol. 47, no. 6, pp. 1646–1653, Jun. 1999.

[9] K. Nishiguchi and M. Kobayashi
Improved algorithm for estimating pulse repetition intervals
*IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 407–421, Apr. 2000.

[10] H. Mardia
New techniques for the deinterleaving of repetitive sequences
In *Proc. IEE Proc. F (Radar Signal Process.)*, 1989, vol. 136, no. 4. IET, pp. 149–154.

[11] D. Milojević and B. Popović
Improved algorithm for the deinterleaving of radar pulses
In *Proc. IEE Proc. F (Radar Signal Process.)*, 1992, vol. 139, no. 1. IET, pp. 98–104.

[12] J. B. Moore and V. Krishnamurthy
Deinterleaving pulse trains using discrete-time stochastic dynamic-linear models
*IEEE Trans. Signal Process.*, vol. 42, no. 11, pp. 3092–3103, Nov. 1994.

[13] T. Conroy and J. B. Moore
The limits of extended Kalman filtering for pulse train deinterleaving
*IEEE Trans. Signal Process.*, vol. 46, no. 12, pp. 3326–3332, Dec. 1998.

[14] A. Logothetis and V. Krishnamurthy
An interval-amplitude algorithm for deinterleaving stochastic pulse train sources
*IEEE Trans. Signal Process.*, vol. 46, no. 5, pp. 1344–1350, May 1998.

[15] N. Visnevski, S. Haykin, V. Krishnamurthy, F. A. Dilkes, and P. Lavoie
Hidden Markov models for radar pulse train analysis in electronic warfare
In *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, vol. 5, pp. v-597–v-600.

[16] J. Liu, H. Meng, Y. Liu, and X. Wang
Deinterleaving pulse trains in unconventional circumstances using multiple hypothesis tracking algorithm
*Signal Process.*, vol. 90, no. 8, pp. 2581–2593, 2010.

[17] J. E. Hopcroft, R. Motwani, and J. D. Ullman
*Introduction to Automata Theory, Languages, and Computation*. Reading, MA, USA: Addison-Wesley, 2001.

[18] N. Chomsky
Three models for the description of language
*IRE Trans. Inf. Theory*, vol. IT-2, no. 3, pp. 113–124, 1956.

[19] N. Chomsky and G. A. Miller
Finite state languages
*Inf. Control*, vol. 1, no. 2, pp. 91–112, 1958.

[20] N. Chomsky
On certain formal properties of grammars
*Inf. Control*, vol. 2, no. 2, pp. 137–167, 1958.

[21] P. Linz
*An Introduction to Formal Languages and Automata*. Boston, MA, USA: Jones & Bartlett, 2016.



**Zhang-Meng Liu** received the Ph.D. degree in statistical signal processing from the National University of Defense Technology (NUDT), Changsha, China, in 2012.

He was a Visiting Scholar with the Big Data and Social Computing Laboratory, led by Prof. Philip S. Yu, in the University of Illinois at Chicago, Chicago, IL, USA, from April 2017 to March 2018. He is now an Associate Professor with the NUDT, working on the interdiscipline of electronics engineering and computer science, especially electronic data mining. He has authored or coauthored more than 30 papers on signal processing, Bayesian learning, and data mining.