IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 1 September 2022, accepted 1 October 2022, date of publication 4 October 2022, date of current version 12 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211938

RESEARCH ARTICLE

Extraction and Sequential Recognition of MFR Pulse Groups in Intercepted Pulse Trains

SHUO YUAN[1], TAO XU[1], MIN ZHANG[1], AND ZHANGMENG LIU[1,2]
[1]School of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China
[2]Tianjin Institute of Advanced Technology, Tianjin 300450, China

Corresponding author: Tao Xu (xutao0474@163.com)

This work was supported in part by the Outstanding Youth Project of Hunan Province under Grant 2020JJ2037, in part by the Young Talents Support Program of Hunan Province under Grant 2019RS2026, and in part by the National Natural Science Foundation of China under Grant 61901494.

ABSTRACT A pulse group is defined as a pulse sequence performing a certain task, such as searching or tracking, in a specific range and azimuth. It serves as a basic information unit of MFR pulse trains. In passive radar data processing, the pulse group structure provides vital support for many applications, such as sequential pulse deinterleaving, pulse group recognition, and working mode identification. In recent years, research related to pulse groups has attracted much attention. In this paper, we propose a pulse group extraction method based on semantic coding and temporal feature clustering. The method can automatically extract pulse group structures from intercepted pulse trains polluted by missing pulses and interferential ones. Furthermore, a temporal pattern recognition method based on the finite state automaton (FSA) is also proposed. Based on extracted pulse group structures, the FSA model achieves sequential recognition of pulse groups in newly intercepted pulse streams. The simulation section verifies the performance of the proposed methods in pulse group extraction and sequential recognition.

INDEX TERMS Multi-functional radar (MFR), pulse group structure extraction, semantic coding model, finite state automaton (FSA), sequential recognition of pulse groups.

I. INTRODUCTION

Multi-functional radar (MFR) may execute numerous functions simultaneously, such as searching and tracking [1], [2]. This kind of radar is widely used in both civil and military fields [3], [4]. A pulse group of an MFR is defined as a fixed arrangement of several pulses, and each pulse group is executed to implement a certain MFR function [5]. Pulse groups contain both values and arrangements of pulse repetition intervals (PRI), and they offer a high-dimensional structure characteristic for deinterleaving [6], [7] and recognition [5], [8], [9], [10], [11], [12], [13] in passive radar signal processing. In this paper, we concentrate on MFRs with fixed parameters and develop a method to automatically extract their pulse group structures based on intercepted pulse trains.

In electronic intelligence (ELINT) systems, several methods have been developed to deinterleave intercepted pulses using their angles of arrival (AOA) and other low-dimensional parameters [14]. A generally used observation set is the pulse description word (PDW), which contains radio frequency (RF), pulse width (PW), and typical PRIs calculated by differentiating the time of arrival (TOA) [15]. PDWs are obtained from intercepted pulse trains and used to match newly arrived pulses to realize radiation source identification in electronic support measure (ESM) systems. These methods may not work for MFRs, since PDWs of MFRs are usually changing rapidly with time. Moreover, PRI values of different radar functions may overlap each other for different MFRs [16]. That is to say, low-dimension PDW parameters do not contain enough information to meet the requirements of pulse deinterleaving and recognition for MFRs [11] in ESM systems.

The pulse group structure presents a high-dimensional feature for distinguishing MFRs in the parameter space. It helps to reduce parameter overlap between different MFRs and has the potential to improve pulse deinterleaving and radar

The associate editor coordinating the review of this manuscript and approving it for publication was Fabrizio Santi.

identification accuracies. Figure 1 provides an overall perspective for the analysis and application of the pulse group structure. Although pulse group structures have widespread applications in the passive processing of radar signals, the way of analyzing and applying them is still an open problem for the ELINT community. In this paper, we mainly address two problems, including the extraction of pulse group structures from preliminarily deinterleaved pulse trains in ELINT systems and the sequential recognition of pulse groups from pulse streams in ESM systems, which have been highlighted in red in Fig. 1(a) and Fig. 1(b) respectively.

The extraction of pulse group structures originates from researches on PRI estimation during pulse deinterleaving. The histogram methods [17], [18], PRI transform method [19], and some other methods [20], [21] have been proposed to extract framework periods of conventional radars. These methods perform satisfyingly for radars having determined PRI modes and values, and they have played a significant role in the past few decades in solving the pulse deinterleaving problem. If the radar does not have stable framework periods, these methods will deteriorate largely in performance. Around 2007, Visnevski et al. pioneered the research in establishing hierarchical models for MFR signals [22], and they also presented a syntactic model to analyze MFR pulse trains in detail [23]. However, the syntactic model is not suitable for the passive analysis of pulse trains since it requires prior knowledge about MFR pulse groups. To automatically analyze the timing pattern in pulse trains, Liu introduced the theories of information and coding to this area and proposed a semantic coding model [24]. This model provides an efficient tool for modeling and analyzing the hierarchical structure of MFR signals. However, the fact that this model ignores real-world noises like interferential pulses and missing ones prevents it from being directly applied to real-world circumstances. In [25], a merging coding strategy was suggested to deal with pulse trains contaminated by data noises and had shown some potential on PRI pattern extraction. However, it was designed for conventional radars originally and is not applicable to the MFR, which has multiple pulse groups.

For another problem of pulse group recognition (also known as radar word recognition in [9], and [10]), Visnevski proposed two methods. The first method is establishing a hidden Markov model (HMM) template of the radar word and applying the Viterbi algorithm to match it with the coding sequence to get the optimal radar word [12]. The second approach is an event-driven method, which can be seen as a simplification of the first one [9]. The two methods obtain good results for radar words with static parameters. However, their implementations rely on empirical parameters, and they may become quite unstable if the parameters are not set properly. In 2010, Liu proposed a method for recognizing MFR pulse groups based on three-level matching, including database level, pulse level, and code sequence level [10]. This approach does not rely on empirical knowledge anymore, but it performs poorly on pulse trains with high missing pulse rates and shows insensitivity to interferential pulses.

In practical applications, intercepted MFR pulse trains are generally contaminated by data noises, but they still adhere to stable hierarchical models controlling the signal-generating actions of the radar. In the hierarchical models, pulse groups form the smallest informational units in pulse trains and generate variegated noise-contaminated intercepted pulse sequences. Therefore, it is believed that the pulse group set provides a simplest dictionary for describing the intercepted pulse trains, or to put it another way, the intercepted pulse trains may be compressed to the largest extent based on the pulse groups, which obeys the Principle of Occam's Razor [26]. This paper seeks to work out a way to reveal the data-model relationship by compressing intercepted pulse trains, which realizes automatic extraction of pulse groups. After that, a finite state automaton (FSA) model is established to recognize pulse groups in newly received pulse streams, which partially verifies the application of pulse group structures.

The main contributions of this paper are three-fold,

(1) A deep insight is given into the compressibility of intercepted MFR pulse trains, and a technique is offered to eliminate the redundancy in pulse trains via semantic coding optimization.

(2) A method based on semantic coding and timing feature clustering is proposed to automatically extract pulse groups from noise-contaminated MFR pulse trains.

(3) A pulse group recognition method based on FSA is proposed, which realizes the sequential recognition of pulse groups in a pulse stream.

II. SEMANTIC CODING MODEL OF PULSE TRAIN AND PULSE GROUP STRUCTURE EXTRACTION

Since the intercepted pulse train is polluted by interferential pulses and missing pulses, the pulse group cannot be extracted intuitively. To solve this problem, this section presents an approach of extracting pulse group structures based on semantic coding and timing feature clustering.

A. DESCRIPTION OF MFR PULSE TRAIN

MFR continuously broadcasts pulse groups into space to detect targets, as shown in Fig. 2. The successive pulse groups correspond to a certain arrangement of different MFR functions, making the pulse train have a hierarchical structure, as shown in Fig. 3. Different pulse groups are constituted by different arrangements of pulses at the bottom layer, and radars emit different pulse groups at the top level according to pre-programmed or situation-dependent operations.

In a pulse group, each pulse can be regarded as a state which is determined by pulse parameters, such as RF and PW [15]. And a pulse group can be described as a transition chain between different states as shown below,

$$q_1 \xrightarrow{PRI_1} q_2 \xrightarrow{PRI_2} q_3 \cdots \xrightarrow{PRI_i} q_i \cdots \xrightarrow{PRI_I} q_I \tag{1}$$

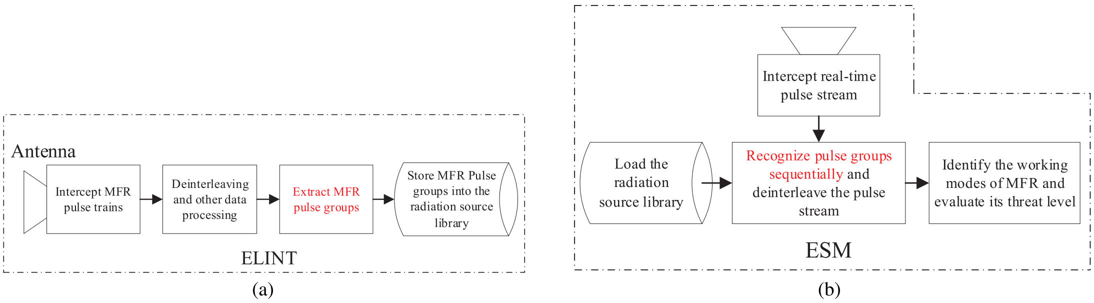

FIGURE 1. MFR pulse group structure extraction and application diagram. (a) Extract pulse group structures in ELINT systems. (b) Recognize pulse groups sequentially in ESM systems.

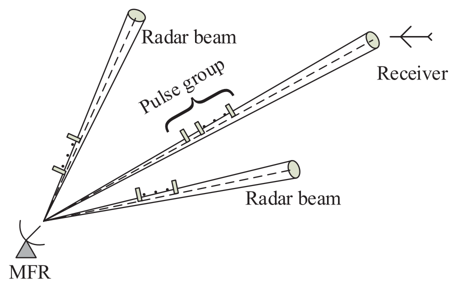

FIGURE 2. Working diagram of multi-functional phased array radar.

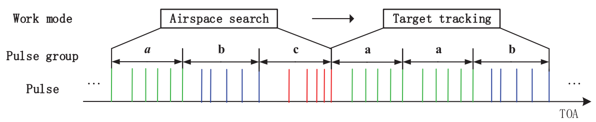

FIGURE 3. Hierarchical structure of the multi-function radar pulse train.

where q_i represents the i-th pulse's state, and PRI_i represents the time interval between the i-th and $(i+1)$-th pulse in this pulse group. For simplicity, pulse parameters except TOA are set to be constant, that is $q_1 = q_2 = \cdots = q_I$. Then, we define $a_i \triangleq q_i \xrightarrow{\text{PRI}_i}$, and the pulse group shown in formula 1 can be expressed as $\boldsymbol{a} = a_1 a_2 \cdots a_I$. The pulse train shown in Fig. 3 is formulated as

$$\begin{aligned} y &= |a_1 \cdots a_5|b_1 \cdots b_5|c_1 \cdots c_5|a_1 \cdots a_5|a_1 \cdots a_5|b_1 \cdots b_5| \\ &= \boldsymbol{abcaab} \end{aligned} \tag{2}$$

In intercepted pulse trains, differential time of arrival (DTOA) sequences can be obtained from the differential calculation of TOAs, but it is usually deviated from the formulation in Eq. 2 due to various noises. The strong directionality of the MFR beam will cause its signal to have a low-interception probability in ESM systems since pulse groups with amplitude lower than the receiver sensitivity will be lost at the receivers. The missing of each pulse group will further destroy the temporal relationship between adjacent pulse groups and significantly increase the number of missing pulses. Moreover, pulses within the pulse group may also be lost with a certain probability due to thermal noise, which is called missing pulse in the pulse group. In addition,

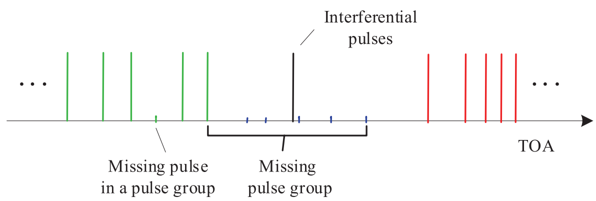

FIGURE 4. Schematic diagram of an intercepted pulse train.

interferential pulses from other radar emitters may also be introduced in the pulse train. The interferential pulses will destroy the structures of pulse groups and the regularity of the pulse train, which further causes great difficulties in the analysis of pulse group structure, and negatively affects the effectiveness and accuracy of the pulse group extraction process. A schematic diagram of an intercepted pulse train is shown in Fig. 4. In the figure, low-amplitude lines represent missing pulses, the solid black line represents an interferential pulse, and high-amplitude lines in colors represent intercepted pulses. The pulse train in Fig. 4 can be formulated as

$$y = |a_1 a_2 \underline{a_3 + a_4} a_5|\delta_1 \delta_2|c_2 \cdots c_5| = \hat{\boldsymbol{a}}\delta_1\delta_2\hat{\boldsymbol{c}} \tag{3}$$

where $\hat{\boldsymbol{a}}, \hat{\boldsymbol{c}}$ are variants of $\boldsymbol{a}$ and $\boldsymbol{c}$ due to missing pulses. δ_1, δ_2 represent the DTOAs between the interferential pulse and its two adjacent pulses before and after it.

By comparing Eq. 2 and Eq. 3, one can easily conclude that the intercepted pulse train (second term of Eq. 3) can be largely compressed (third term of Eq. 3). However, if the prior knowledge about the pulse group structures is absent, which is the real case in most ESM systems, the boundaries between pulse groups in Fig.4 become very blurred, and it will be very difficult to compress the pulse trains. In this section, we will provide a deep insight into the compressibility of pulse trains, and work out a way to extract pulse groups from intercepted MFR pulse trains contaminated by pulse noises.

B. A SCHEMATIC INTRODUCTION OF THE WHOLE ALGORITHM

The pulse group extraction is to extract the hidden pulse group structures from noise-contaminated MFR pulse trains

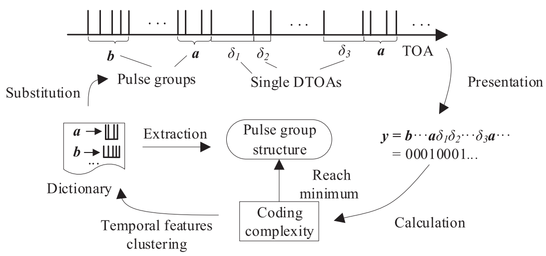

FIGURE 5. Schematic diagram of pulse group structure extraction.

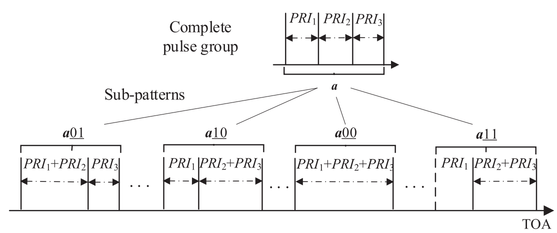

FIGURE 6. An merging coding example.

without any prior information, which is a demanding data mining problem. In this part, we propose a method based on semantic coding and temporal feature clustering to compress the pulse train and extract the pulse group structures. The method divides into three procedures, which are pulse train modeling from the perspective of semantic coding, temporal feature clustering and pulse group structure extraction. The three procedures will be detailed in the following three parts. The diagram of extracting pulse group structures is shown in Fig. 5.

At first, a semantic coding model is established to formulate pulse trains. The model contains three components, including single-pulse coding, pulse group coding and dictionary coding. The coding complexity is also defined to measure the conciseness of a certain coding strategy [27]. Based on the model, the pulse train is described as a binary sequence to evaluate how a pulse group set performs in encoding a pulse train. Secondly, we establish coding optimization rules to cluster frequent pulse subsequences and compress the pulse train. According to Occam's razor principle (also known as the simplest criterion) [26], the true MFR pulse group set corresponds to the frequent pulse subsequence set that compresses the pulse train most. That is, the pulse groups stored in the semantic coding dictionary provide a reconstruction of the MFR pulse patterns when the coding complexity reaches the minimum. Therefore, in this procedure, we extract MFR pulse groups by iteratively reducing the semantic coding complexity of the pulse train until a minimum is reached. Thirdly, a clustering method is applied to clarify true pulse groups from the semantic coding dictionary which contains incomplete patterns.

C. SEMANTIC CODING MODEL OF THE PULSE TRAIN

In this part, a semantic coding model is established to describe the pulse train as a binary coding sequence. The total coding complexity is defined as the sum of the complexities of single pulse coding, pulse group coding and dictionary coding, i.e.,

$$c = c_s + c_{pg} + c_d \tag{4}$$

where the coding complexity of single pulses c_s is defined as

$$c_s = N_{res} \times c_0 \tag{5}$$

where N_{res} represents the number of single pulses encoded in the pulse train. $c_0 = log_2(DTOA_{max}/\delta) + 1$ represents the code length of a single pulse, where $DTOA_{max}$ represents the preset upper limit of the DTOA range; δrepresents the quantization unit for DTOA quantization and encoding; "1" represents the prefix symbol of codes to distinguish between pulse group and single pulse in the binary coding sequence. For example, if one set $DTOA_{max} = 1\ ms$, and $\delta = 1\ us$, then the code length of a single DTOA is $c_0 = log_2(1\ ms/1\ us) + 1 = 11$. Assuming DTOA = 100us, the prefix coding symbol of a single pulse is 0, and the coding sequence of this pulse is 0|0001100100.

In Fig. 4, missing pulses divide into two categories, one is pulses within a pulse group and the other is pulse groups missed as a whole. The former pulses have no effect on other pulse groups. However, if a pulse group is missed as a whole, the structure of the subsequent pulse group is also destroyed. Due to the low-interception probability of MFR signals, pulse groups are lost with a high probability in the intercepted pulse train. These missing pulses bring great challenges to pulse group coding. To enhance the significance of pulse group structures in the intercepted pulse train, it is necessary to code incomplete pulse groups containing missing pulses according to their complete counterparts, which is called merging coding in this paper [25]. The merging coding of a three-PRI pulse group is shown in Fig. 6.

Assuming a pulse group with n PRIs, there are $n+1$ pulses and the number of internal pulses is $n-1$. If internal pulses are arbitrarily lost, the number of possible cases is $2^{n-1} - 1$ (The subtracted 1 indicates no missing pulse). In view of the importance of the first PRI, when the previous pulse group is lost, remaining pulses of the current pulse group are required to be complete for merge coding. The number of such case is 1, so there is a total of 2^{n-1} cases called sub-patterns. If this n-PRI pulse group is coded as $\boldsymbol{a}$, n-1 coding bits are supplemented after $\boldsymbol{a}$, and 2^{n-1} generated cases indicate the corresponding sub-patterns.

After the merge coding, the coding complexity of pulse groups is expressed as,

$$c_{pg} = \left(\sum_{j=1}^{J} N_j\right) \times \left(\sum_{j=1}^{J} \frac{N_j}{N_{sum}} log_2(N_j/N_{sum}) + 2\right) + \sum_{j=1}^{J}\sum_{i=1}^{I} n_{ij} \cdot b_j \tag{6}$$

where $N_{sum} = \sum_{j=1}^{J} N_j$ is the total number of pulse groups in the pulse train. $\sum_{j=1}^{J} \frac{N_j}{N_{sum}} log_2(N_j/N_{sum}) + 2$ indicates the average coding length of each pulse group [28]. "2" = "1 + 1" with the first "1" corresponding to the code type prefix, which is used to distinguish between pulse group codes and single pulse codes, while the second "1" is a missing pulse indicator used to indicate whether the pulse group contains missing pulses; n_{ij} represents the occurrence time of the i-th sub-pattern of the j-th dictionary element in the pulse train, and b_j represents the supplementary bits of the j-th dictionary element.

The dictionary complexity c_d depends on the number of PRIs stored in the dictionary [24],

$$c_d = \left(\sum_{j=1}^{J} I_j \right) \times c_0 \tag{7}$$

where I_j represents the PRI number of the i-th element in the dictionary, and J represents the total number of elements.

D. TIMING FEATURE CLUSTERING BASED ON SEMANTIC CODING MODEL

At the beginning of a pulse train coding process, all pulses in the pulse train are coded as single pulses. The coding complexity can be reduced by establishing a dictionary of repeated pulse structures in the pulse train. Aiming at minimizing the coding complexity, four steps are followed to construct a dictionary, which are frequent item extraction, frequent item coding into a dictionary, dictionary element concatenation, and redundant dictionary element deletion. Procedures of the four steps are described below briefly to enhance the conciseness and completeness of the paper, and we refer readers to our previous paper [25] for more details.

Step 1: Frequent itemset extraction. Despite the fact that pulse noises destroy the pulse group structure, the individual PRIs comprising pulse groups have a significant frequency in the pulse train, which are frequent DTOAs. A frequent DTOA set can be obtained by numerically clustering the DTOAs [29]. Each frequent item contains four components: DTOA mean value $DTOA_i$, occurrence frequency M_i ($M_i > M_0$, M_0 is a frequency threshold), sequence index set of head pulses $\{h_m^{(i)}\}_{m=1,\cdots,M_i}$, and tail pulse index set $\{t_m^{(i)}\}_{m=1,\cdots,M_i}$ of the frequent items,

$$\left\{ DTOA_i, M_i, \{h_m^{(i)}\}_{m=1,\cdots,M_i}, \{t_m^{(i)}\}_{m=1,\cdots M_i} \right\}_{i=1,\cdots,I} \tag{8}$$

where I denotes the number of frequent items. The frequent item extraction can provide preprocessing for dictionary construction and eliminate the influence of interferential pulses.

Step 2: Frequent item coding into a dictionary. In this step, each frequent item $DTOA_i$ is tentatively encoded into a dictionary with a changing coding complexity $\Delta c_1^{(i)}$. Finally, the frequent item that reduces the coding complexity of the pulse train most is retained. This step is iterated until the complexity does not reduce any more.

Step 3: Dictionary element concatenation. To get a high-dimensional pulse group structure, the single DTOAs need to be concatenated to construct higher-dimensional pulse groups. Any two dictionary elements D_i and D_j are taken as candidates for concatenation, and the effectiveness of each concatenation is evaluated coarsely according to the overlapping of their head and tail pulses and accurately to the coding complexity decrease $\Delta c_2^{(i,j)}$ after connection. The concatenation leading to the largest decrease is taken, and this step is terminated when the coding complexity no longer decreases.

Step 4: Redundant dictionary element deletion. After concatenating the dictionary elements, many shorter patterns may be left over in the dictionary due to data noises. These elements should be deleted to avoid repetition. As the encoding process in step 1, we attempt to delete each dictionary element D_i according to the complexity change $\Delta c_3^{(i)}$. By collating the pulse group dictionary, we finally retain dictionary elements that encode the pulse train with a minimal complexity.

E. EXTRACTION OF PULSE GROUP STRUCTURE BASED ON DICTIONARY ELEMENT CLUSTERING

The restoration degree of pulse groups is improved after encoding complete pulse groups and incomplete pulse groups together as described in merge coding. Furthermore, frequent item extraction reduces the incompleteness in the pulse group structure caused by interferential pulses. However, since the lack of prior information in passive analysis problem, partial structures of the complete pulse group still appear in the dictionary. For example, $\boldsymbol{a'} = a_2a_3 \cdots a_{end}$ is a partial structure of $\boldsymbol{a} = a_1a_2a_3 \cdots a_{end}$. Therefore, final pulse groups need to be extracted from the dictionary set.

The pulse train coding strategy can be optimized by encoding true MFR pulse groups and their variants caused by data noises together as described in merge coding. And merge coding also helps to reduce the incompleteness of the pulse group structure caused by interferential pulses. However, in spite of the great efforts made to normalize the elements in the dictionary, incomplete pulse groups may still be retained in the dictionary. For example, $\boldsymbol{a'} = a_2a_3 \cdots a_{end}$ is an incomplete variant of $\boldsymbol{a} = a_1a_2a_3 \cdots a_{end}$. Therefore, an extra clustering procedure is required to extract final pulse groups from the automatically constructed dictionary.

Noting that complete patterns usually have significant structural similarities with their incomplete variants, we propose a clustering method to distinguish incomplete pulse groups and remove them from the dictionary. The clustering process is sketched in Fig. 7 and stated in detail as follows.

Step 1: Clustering patterns according to structural similarity. The clustering criterion is that pulse group structures in the same category can convert between each other via a small number of element extensions or deletions. For example, $\{a_1, a_2, a_3\}$ is an extended pattern of $\{a_1, a_2\}$, on the contrary, the latter is a partial pattern.

Step 2: Obtaining cluster centers as pulse group structures. Summing the occurrence times of the pulse groups in each

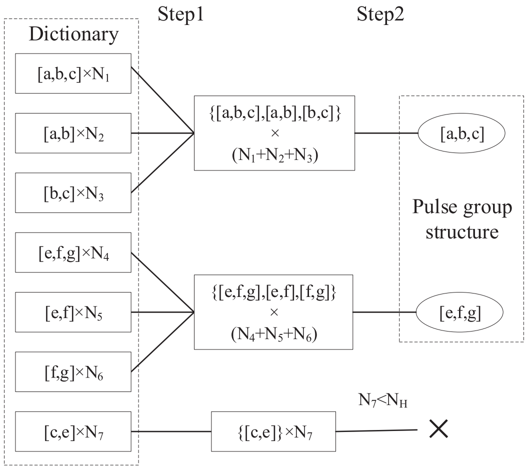

FIGURE 7. Schematic diagram of extracting pulse group structure from dictionary.

category as its total frequency and setting the frequency threshold N_H. Categories with frequencies lower than the threshold are removed as outliers, and the other categories are retained as pulse group structures.

By combining the three procedures of constructing the semantic coding model, optimizing the dictionary set, and extracting final pulse group structures from the dictionary, the proposed method can extract MFR pulse group structures from intercepted pulse trains polluted by pulse noises. The extracted pulse groups provide prior information for subsequent data processing applications, such as sequential pulse deinterleaving [6] and online pulse group recognition [8], [9], [10], [11] in electronic support systems.

III. SEQUENTIAL RECOGNITION OF PULSE GROUPS BASED ON FINITE STATE AUTOMATON

Sequential pulse group recognition is to recognize the pulses that conform to specific parameters and variation rules along the time axis. This section introduces a syntactic pattern recognition tool, the finite state automaton (FSA), and proposes a sequential recognition algorithm for MFR pulse streams.

FSA is a mathematical model that describes finite states, transitions and actions between states. It has a limited set of inputs, and each state can be transferred to zero or more other states by the input string [31]. If we take pulses as different states and time intervals as inputs, the pulse train analysis along time can be treated as state transitions of FSA. The m-th pulse state in the i-th pulse group is expressed as $q_{i,m}$, and the pulse group structure of the -th pulse group is $\{PRI_{i,1}, PRI_{i,2}, \cdots PRI_{i,end}\}$, then a pulse train can be expressed as:

$$q_{1,1} \xrightarrow{PRI_{1,1}} q_{1,2} \xrightarrow{PRI_{1,2}} q_{1,3} \cdots q_{1,end} \xrightarrow{PRI_{1,end}} q_{2,1} \cdots \tag{9}$$

A. FSA MODEL FOR MFR PULSE GROUP RECOGNITION

FSA is modeled by a five tuple [30]:

$$Z = (Q, \Sigma, \delta, q_0, F) \tag{10}$$

where

Q represents all possible states of the automaton, which can be determined by the pulse group number i and the pulse position m in this pulse group, that is, $Q = \{q_{1,1}, q_{1,2}, \cdots q_{i,m}, \cdots, q_{I,M_I}\}$, where I indicates the total number of MFR pulse groups. M_I represents the total number of PRIs in the I-th pulse group;

$\Sigma = \{DTOA(q_{i,m}, q_{j,n}), 1 \leqslant i, j \geqslant I, n \geqslant M_i\}$ represents DTOAs of state transitions, corresponding to the input set of the FSA;

$q_0 \in Q$ represents the starting state of the FSA;

$F = \$$ indicates the terminating state of the FSA indicating that the recognition is completed or the FSA state is not updated for a long time.

δ represents the state transition function which refers to DTOA conditions that need to be met for state transition. The state transition in the intercepted pulse train can be classified into two cases: the state transition in a pulse group and state transition across pulse groups. Tables 1 and 2 are state transition tables within a pulse group and crossing pulse groups, respectively. The Eq. 11 and Eq. 12 are state transition functions for two cases respectively.

$$\delta(q_m, q_n) = \sum_{k=m}^{n-1} PRI_{i,k}, \quad \delta(q_m, q_n) \leq \sum_{k=m}^{M_i} PRI_{i,k} \tag{11}$$

$$\delta(q_{i,m}, q_{j,n}) = \sum_{k=m}^{M_i} PRI_{i,k} + T + \sum_{k=1}^{n-1} PRI_{j,k},$$
$$\delta(q_{i,m}, q_{j,n}) > \sum_{k=m}^{M_i} PRI_{i,k} \tag{12}$$

where T represents the specific DTOA caused by missing pulse groups during the state transition across pulse groups. When there is no missing pulse group, $T = 0$, otherwise,

$$T = a_1 \cdot \mathrm{T}_1 + a_2 \cdot \mathrm{T}_2 + \cdots + a_i \cdot \mathrm{T}_i + \cdots + a_I \cdot \mathrm{T}_I \tag{13}$$

where a_i is the number of missing i-th MFR pulse groups, and T_i is the framework period of the i-th pulse group [6],

$$T_i = \sum_{m=1}^{M_i} PRI_{i,m} \tag{14}$$

where M_i represents the number of PRIs in the i-th pulse group. It can be seen that T is a linear combination of framework periods of MFR pulse groups when it is not zero.

The model of FSA is shown in Fig. 8. It includes two basic structures: the input pulse train and the finite state controller (FSC). The current state and state transition rules are stored in the FSC. A reading head is controlled by FSC to read the TOA of pulses in the pulse train from left to right.

In Fig. 8, after the FSC updates its status at t_{i+3}, the starting point of the current pulse group is t_i, and the sequence number

TABLE 1. Table of state transitions within a pulse group.

s \ 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		0	PRI_2	$\cdots$	$PRI_2 + \cdots + PRI_{M-1}$
q_3			0	$\cdots$	$PRI_3 + \cdots + PRI_{M-1}$
$\cdots$				$\cdots$	$\cdots$
q_M				$\cdots$	0

TABLE 2. Table of state transitions crossing pulse groups.

s \ s'	$q_{j,1}$	$q_{j,2}$	$\cdots$	$q_{j,N}$
$q_{i,1}$	$\delta(q_{i,1}, q_{j,1})$	$\delta(q_{i,1}, q_{j,2})$	$\cdots$	$\delta(q_{i,1}, q_{j,N})$
$q_{i,2}$	$\delta(q_{i,2}, q_{j,1})$	$\delta(q_{i,2}, q_{j,2})$	$\cdots$	$\delta(q_{i,2}, q_{j,N})$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$q_{i,M}$	$\delta(q_{i,M}, q_{j,1})$	$\delta(q_{i,M}, q_{j,2})$	$\cdots$	$\delta(q_{i,M}, q_{j,N})$

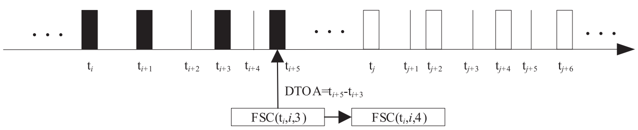

FIGURE 8. **Physical model of automata.**

of this pulse group is i. When FSC reads the pulse at time t_{i+4}, the input DTOA is $t_{i+4} - t_{i+3}$, which does not meet the state transition function, so the FSA state doesn't change. Next, FSC reads the pulse at time t_{i+5}, and the input DTOA is $t_{i+5} - t_{i+3}$ and it satisfies the state transition function in a pulse group, so the FSA updates its state to the fourth pulse of the i-th pulse group. When new pulses arrive sequentially, the FSA reads them one by one to determine their states, which in fact realizes sequential pulse group recognition. The FSA state is represented as,

$$s = [t_s, i, m] \tag{15}$$

where t_s indicates the start time of the current pulse group, represents the index of this kind of pulse group, and m is the location index of the current pulse in the pulse group. The start time t_s and the pulse group index i jointly determine the pulse group where the FSC is located, and the pulse index m determines the specific position of FSA.

B. SEQUENTIAL RECOGNITION OF MFR PULSE GROUPS

Based on the above FSA model, this section proposes a sequential recognition algorithm for MFR pulse groups, and its pseudo-codes are listed as Algorithm 1 and Algorithm 2.

When a new pulse is intercepted, the FSA is judged whether it has been started or not, and the algorithm is divided into FSA startup and update processes accordingly.

If the FSA has not been started, the intercepted pulse's TOA is accumulated to the pulse set $\boldsymbol{T_s}$ for further starting judgement. Then, $\boldsymbol{T_s}$ is input into the FSA startup algorithm along with the startup template $\boldsymbol{F}$ to determine whether the FSA should be started. The startup template is shown in Table 3. The pulses in $\boldsymbol{T_s}$ are successively taken as the start point, and a DTOA sequence is obtained to match with the startup template. The FSA would start when the number of matching DTOAs exceeds N_T. Once the FSA has been started, its state is updated and $\boldsymbol{T_s}$ is cleared for restarting. The threshold for the number of matched DTOAs is

$$N_{T_i} = k \cdot length(F_i) \tag{16}$$

where $k \in [0, 1]$, which means that the number of matching DTOAs must be k times the number of PRIs in the i-th matching template.

When the FSA has been started, a differential time TOA_c of each newly intercepted pulse is obtained by computing the minus of the pulse arrival time $DTOA_c$ and the FSA time. Then three evaluation operations will be performed on $DTOA_c$ to implement pulse recognition.

1) If $DTOA_c$ is smaller than the remaining time of the current pulse group, it is evaluated as a state transition within this pulse group. Then match $DTOA_c$ with the state transferring rules in Table 1, if a certain rule is satisfied, one should determine the pulse group's position where the new pulse is located, and update the FSA state accordingly. If $DTOA_c$ does not obey any state transferring rule in Table 1, this pulse is judged as an interferential pulse.

3)If $DTOA_c$ is larger than the remaining time of the current pulse group $\sum_{k=m}^{M} PRI_{i,k}$ but smaller than the restart threshold T_H, it is preliminarily judged as a state transition crossing different pulse groups, and $DTOA_c$ should be prejudged according to rules in Table 2. Here, the integer $a_1, a_2, \cdots, a_I, j, n$ needs to be determined according to Eq. 12 and Eq. 13. If no valid solution can be obtained, the pulse is regarded as an interferential one. Otherwise, the current pulse's possible state is obtained. The term "possible" here means that even if there is a solution, the pulse could still be an interferential pulse or a wrong solution. If the solution pulse exists, and the new pulse group's starting time and sequence are saved in the pulse set $\boldsymbol{P}$ to be recognized. If the number of pulses belonging to the i-th pulse group in the set $\boldsymbol{P}$ exceeds N_{T_i}, it is regarded that the possible location

TABLE 3. Automata startup template.

number	Startup template
F_1	$\{0,\ PRI_{1,1},\ PRI_{1,1}+PRI_{1,2},\ \cdots,\ PRI_{1,1}+PRI_{1,2}+,\cdots,+PRI_{1,M_1}\}$
F_2	$\{0,\ PRI_{2,1},\ PRI_{2,1}+PRI_{2,2},\ \cdots,\ PRI_{2,1}+PRI_{2,2}+,\cdots,+PRI_{2,M_2}\}$
$\cdots$	$\cdots$
F_I	$\{0,\ PRI_{I,1},\ PRI_{I,1}+PRI_{I,2},\ \cdots,\ PRI_{I,1}+PRI_{I,2}+,\cdots,+PRI_{I,M_I}\}$

Algorithm 1 Sequential Recognition Algorithm Based on Automata

Input: Automata's state $S = \{t_s, i, m\}$; TOA of current pulse TOA_c; Pulse set to be started $\boldsymbol{T_s} = \{t_1, t_2, \ldots, t_n\}$; Pulse set to be recognized $\boldsymbol{P} = \{p_1, p_2, \ldots, p_L\}$; Startup mark of automata S_F; Startup template $\boldsymbol{F} = \{F_1, F_2, \ldots, F_I\}$; Restart DTOA threshold T_H; State transition Table 1 and 2;

if $S_F == 0$ **then**
  $\boldsymbol{T_s} \Leftarrow \{t_1, t_2, \ldots, t_n, TOA_c\}$
  $S, S_F = Start(\boldsymbol{T_s}, \boldsymbol{F})$
  if $S_F == 1$ **then**
    Empty the impulse train set to be started: $\boldsymbol{T_s} \Leftarrow \varnothing$
  end if
else
  $DTOA_c \Leftarrow TOA_c - (ts+\text{Table1}(m, M_i))$;
  if $DTOA_c < \sum_{k=m}^{M_i} PRI_{i,k}$: The new pulse does not exceed the current pulse group **then**
    if $DTOA_c ==\text{Table1}(m, n)$ **then**
      Match successfully. Update $S = \{t_s, m, n\}$ and pulse recognition result
    else
      Treat the current pulse as an interferential pulse and input a next pulse
    end if
  else $\{DTOA_c > \sum_{k=m}^{M} PRI_{i,k}$: The new pulse exceeds the current pulse group$\}$
    if $DTOA_c >= T_H$ **then**
      $S_F \Leftarrow 0$;
    else
      if $DTOA_c ==$ Table2$((i, m), (j, n))$ **then**
        Match successfully and add the current pulse to the pulses set $\boldsymbol{P}$ to be recognized
        $N_T \Leftarrow 0.5 \cdot length($ Table2$((i, 1), (j, :)))$
        if There are more than or equal to N_T pulses in $\boldsymbol{P}$ belonging to the same pulse group **then**
          Update S and pulse recognition results
        end if
      else
        The current pulse is regarded as an interferential pulse
      end if
    end if
  end if
end if

Output: Recognition results

Algorithm 2 Automata Startup Algorithm

Input: The pulse trains set to be started $\boldsymbol{T_s} = \{t_1, t_2, \ldots, t_n\}$; Startup template $\boldsymbol{F} = \{F_1, F_2, \ldots, F_I\}$;

for k = 1,2,...,n-1 **do**
  $L \Leftarrow \{t_k, t_k + 1, \ldots, t_n\}$
  $L \Leftarrow L - t_k$;
  for F_i *in* $\boldsymbol{F}$ **do**
    if $|F_i \cap L| >= N_{T_i}$ **then**
      $S_F \Leftarrow 1$
      Update the position of the last matching pulse $S = \{t_k, i, F_i\}$
    end if
  end for
end for

Output: S, StartFlag

of these pulses is true, and the FSA state is updated according to the latest pulse.

3) If $DTOA_c > T_H$, it indicates that there have been no pulses for a long time. Then one should terminate this FSA and turn to step 1 for pulse accumulation and wait the FSA to restart.

The computational complexity of the pulse group recognition algorithm based on FSA can be divided into two parts of FSA startup (Algorithm 2) and FSA matching (Algorithm 1). Assuming that there are n pulses in the pulse set $\boldsymbol{T_s}$ to be started when the FSA starts, its computational complexity is proportional to $O((n-1)I)$, where I is the number of startup templates in $\boldsymbol{F}$. In the process of FSA matching, each pulse experiences once match as shown in Algorithm 1. If the trial solution of Eq. 11 and Eq. 12 when state transfers across pulse groups is not considered, the computational complexity of FSA matching is proportional to $O(L)$, where L is the number of pulses in the intercepted pulse train.

IV. SIMULATION

In this section, simulations are carried out to verify the performance of the pulse group extraction and recognition methods proposed in this paper. In the simulations, the effects of real-world factors, including missing pulses, interferential pulses, number of intercepted pulse groups, are investigated. This section is divided into three parts. The first part introduces the parameter settings. The second part evaluates the influence of pulse noises on pulse group extraction based on semantic coding, and the third part shows the performance of the sequential pulse group recognition based on FSA.

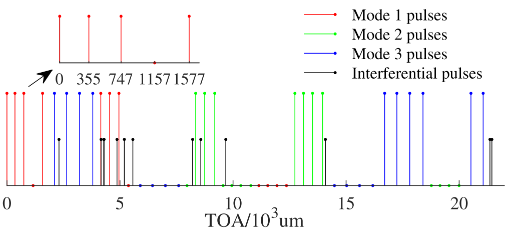

FIGURE 9. Simulation diagram of an intercepted pulse train.

A. SIMULATION PARAMETER SETTINGS

An MFR pulse train with three kinds of pulse groups is simulated, and the PRI vectors of pulse groups transmitted by MFR are [355, 392, 410, 420] *us*, [371, 382, 399, 444] *us* and [525, 540, 570, 585] *us* respectively. The standard deviation of TOA measurement is 0.2 *us*.

The conversion probability matrix between the radar pulse groups is:

$$T = \begin{bmatrix} 0.3 & 0.4 & 0.3 \\ 0.45 & 0.3 & 0.25 \\ 0.3 & 0.3 & 0.4 \end{bmatrix} \tag{17}$$

Based on the aforementioned pulse train setting parameters, the simulation of a partial pulse train is displayed in Fig. 9 when the probability of missing pulse group (PMPG) is set to 0.4, the probability of missing pulse in a pulse group (PMPIG) is 0.2, and the interferential pulse ratio (IPR) is 0.6.

In the simulation of pulse group extraction, the PMPIG is fixed to 0.2, the threshold of frequent item extracting is set to 0.15 times of the highest DTOA frequency in the pulse train, and the maximum allowable DTOA is 2000 *us*. The quantization unit for the coding process is $\delta = 1$ *us*. The threshold N_H for removing discrete points during clustering is set to 0.1 of the number of intercepted pulse groups. The extraction accuracy is used to assess the extraction performance. Assuming the number of simulations is S and the number of successful pulse group extraction is PG_{ext}, the extraction accuracy is

$$Extraction\ accuracy = \frac{PG_{ext}}{S} \tag{18}$$

The criterion for successful extraction is to extract all pulse groups in the pulse train, and the PRI error is less than 0.5 *us*.

In the simulation of sequential pulse group recognition, the number of intercepted pulse groups N is fixed at 200, and the proportion of matching threshold in Eq. 16 is $k = 0.5$. The restart threshold is $T_H = 2 \times 10^4$ *us*. If the number of simulations is S, and the successful pulse group recognition time of the i-th simulation is PG_{reg_i}, the recognition accuracy is

$$Recognition\ accuracy = \frac{\sum_{i=1}^{S} PG_{reg_i}}{S \times N} \tag{19}$$

The effectiveness of the recognition is tested by comparing the starting time and sequence number of the FSA-recognized pulse group with the targeted pulse group.

B. SIMULATION: INFLUENCE OF PULSE NOISES ON PULSE GROUP EXTRACTION METHOD BASED ON SEMANTIC CODING

Set the number of intercepted pulse groups as $N = 200$ and 300. Firstly, the influence of PMPG on extraction is investigated when the IPR is fixed at 0. Then the PMPG is fixed at 0.4 to investigate how IPR affects extraction. Simulating 100 times yields the extraction accuracy.

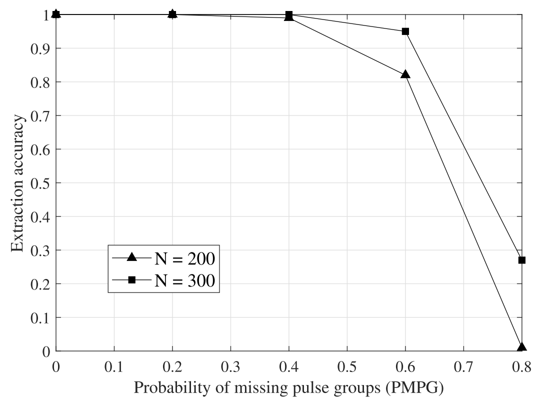

FIGURE 10. Effect of PMPG on extraction.

It can be seen from results in Fig. 10 that when there is no interferential pulse, the extraction accuracy decreases with the PMPG increases. When 200 pulse groups are intercepted and the PMPG is lower than 0.4, the semantic coding method has a perfect performance; when the PMPG is 0.6, the extraction accuracy is 82%; and when the PMPG reaches 0.8, it loses its extraction ability. This abrupt decline is because incomplete pulse groups have a significant frequency and are mistaken for complete pulse groups. The extraction effect is improved by intercepting 300 pulse groups, and at a PMPG of 0.6, the extraction accuracy can still be as high as 96%.

The results in Fig. 11 show that when 300 pulse groups are intercepted and the IPR does not exceed 1, the semantic coding method can accurately extract pulse group structures. When the IPR reaches 1.2, the extraction accuracy deteriorates seriously from 97% to 64%, which is because too many interferential pulses destroy true MFR frequent terms, resulting in incompleteness in pulse group structures. When the number of intercepted pulse groups is 200, the extraction effect deteriorates more seriously from 82% to 36% when the IPR reaches 1.2.

It can be seen from two figures that when 300 pulse groups are intercepted, the maximum missing pulse group probability and the maximum interferential pulse ratio that the method can tolerate are 0.6 and 1, respectively. In addition, the more pulse groups intercepted, the better the extraction performance is.

C. SIMULATION: INFLUENCE OF PULSE NOISES ON AUTOMATON-BASED PULSE GROUP RECOGNITION

This simulation part is set up to compare the pulse group recognition accuracies of the proposed method and two other

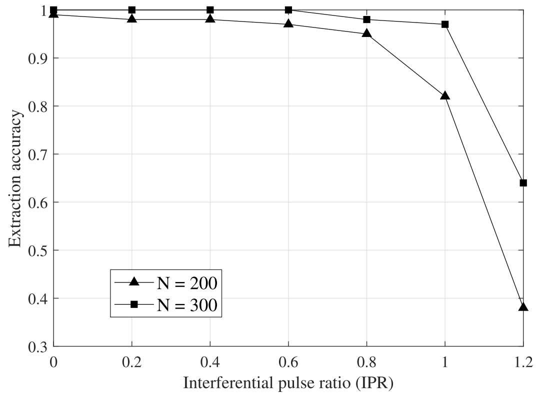

FIGURE 11. Effect of IPR on extraction.

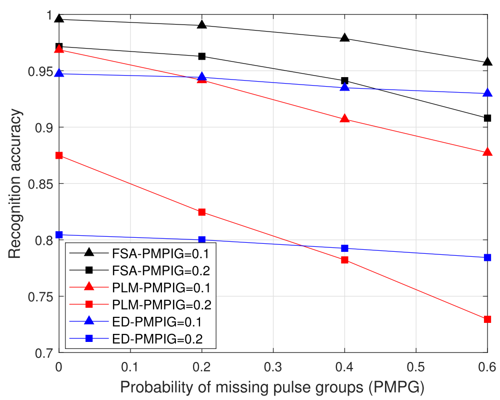

FIGURE 12. Effect of PMPG on recognition.

existing ones, including the event-driven (ED) algorithm in [9] and the pulse-level matching algorithm (PLM) in [10]. The PMPIP is set to 0.1 and 0.2. We investigate the effect of PMPG on recognition accuracy when the IPR is fixed at 1 and the effect of IPR on recognition when PMPG is fixed at 0.4. The recognition accuracy is calculated by simulating 100 times. It should be noted that the empirical IPR required by the ED algorithm must be less than 1. Therefore, when we generate data with IPR = 1, the empirical IPR of the algorithm is set to 0.99.

As can be seen in Fig. 12 and Fig. 13, the FSA algorithm has great advantages over the other two existing algorithms under all considered scenarios. The pulse noise is most significant when PMPG is 0.6, IPR is 1, and PMPIG is 0.2 in Fig. 12. In this scenario, the FSA algorithm still maintains a recognition accuracy of 91%, while the other two methods are less than 80%. It can be seen from Fig. 12 that the recognition accuracies of the FSA algorithm and the PLM algorithm decrease gradually when the PMPG increases from 0 to 0.6. Although the ED algorithm does not achieve a recognition accuracy as high as the FSA algorithm, it is not sensitive to PMPG, with a decrease of only 1%. This is mainly because the empirical parameters of the algorithm do not rely on PMPG, so its performance is only slightly affected by this factor [9]. In addition, it can be concluded from Fig. 13 that,

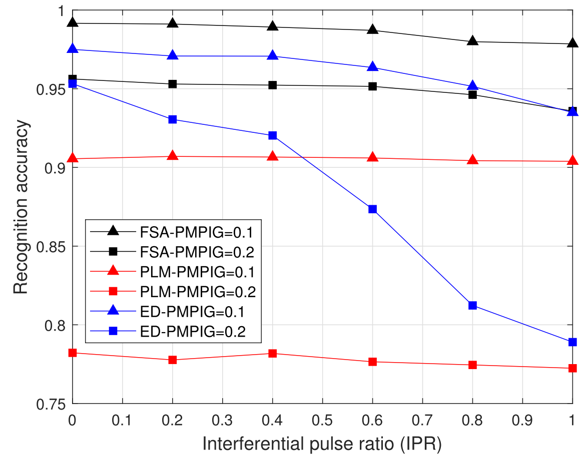

FIGURE 13. Effect of IPR on recognition.

both FSA and PLM algorithms are insensitive to the IPR factor. This is because interferential pulses are generally skipped over and only true radar pulses are considered for recognition. According to the results in Fig.12 and Fig.13, when PMPIG increases from 0.1 to 0.2, the recognition accuracies of FSA, PLM and ED algorithms reduce by 2%-5%, 10%-15% and 2%-15% respectively, which further indicates the robustness of the FSA algorithm to data noises.

V. CONCLUSION

This paper addresses the problems of automatic extraction and sequential recognition of MFR pulse groups, and two methods based on semantic coding and finite state automaton have been proposed respectively. The simulation results show that the proposed methods have good performance in the considered scenarios.

For extracting MFR pulse groups from noise-contaminated pulse trains, an extraction method based on semantic coding and temporal feature clustering is proposed. This method utilizes coding theory and Occam's razor principle to realize automatic extraction of MFR pulse groups successfully in spite of data noises. If more than 300 pulse groups are intercepted, our method can always achieve an extraction accuracy not lower than 95%, even if the ratios of missing pulse groups and interferential pulses reach as high as 0.6 and 1, respectively.

For sequentially recognizing MFR pulse groups based on the extracted pulse group structure, we compared our FSA-based method with two other methods. Simulation results show that our method surpasses its counterparts with very large performance margins, its recognition accuracy exceeds 91% in all the simulations.

REFERENCES

[1] L. Jian-Bin, H. Wei-Dong, and Y. Wen-Xian, "Resource management algorithms based on covariance control for phased array radars," *Acta Electronica Sinica*, vol. 35, no. 3, pp. 402–408, 2007.

[2] S. Sergio and M. Tarantino, *Multifunction Array Radar: System Design and Analysis*. Norwood, MA, USA: Artech House, 1994, pp. 213–245.

[3] M. A. Richards, J. A. Scheer, and W. A. Holm, *Principles of Modern Radar: Basic Principles*. U.K.: IET Digital Library, 2010.

[4] W. L. Melvin and J. A. Scheer, *Principles of Modern Radar: Advanced Techniques*. U.K.: IET Digital Library, 2013.

[5] O. Jian, "Research on behavior recognition and prediction techniques against multi-function radar," Ph.D. thesis, Dept. Inf. Commun. Eng., National Univ. Defense Technol., China, 2017.

[6] Z.-M. Liu, "Online pulse deinterleaving with finite automata," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 2, pp. 1139–1147, Apr. 2020.

[7] W. Cheng, Q. Zhang, J. Dong, C. Wang, X. Liu, and G. Fang, "An enhanced algorithm for deinterleaving mixed radar signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 3927–3940, Dec. 2021, doi: 10.1109/TAES.2021.3087832.

[8] S. Kang, S. Yuan, C. Wang, and Z. Liu, "Sequential extraction and recognition of pulse group structure for multi-function radar," *IET Radar, Sonar Navigat.*, vol. 16, no. 4, pp. 678–691, Apr. 2022, doi: 10.1049/rsn2.12212.

[9] N. Visnevski, "Syntactic modeling of multi-function radars," Ph.D. thesis, Dept. Elect. Comput. Eng., McMaster Univ., Hamilton, ON, Canada, 2005.

[10] H.-J. Liu, "Researches on identification key technology for radar emitter," Ph.D. thesis, Dept. Inf. Commun. Eng., Nat. Univ. Defense Technol., China, 2010.

[11] 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. (ICASSP)*, Philadelphia, PA, USA, Mar. 2005, pp. 597–600.

[12] Z.-M. Liu, "Recognition of multifunction radars via hierarchically mining and exploiting pulse group patterns," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 6, pp. 4659–4672, Dec. 2020.

[13] X. Li, Z. Liu, and Z. Huang, "Attention-based radar PRI modulation recognition with recurrent neural networks," *IEEE Access*, vol. 8, pp. 57426–57436, 2020, doi: 10.1109/ACCESS.2020.2982654.

[14] S. Lin and S. J. C. D. Sciortino, "Comparison of time of arrival vs. multiple parameter based radar pulse train deinterleavers," *Proc. SPIE*, vol. 6235, May 2006, Art. no. 62351K.

[15] R. G. Wiley, *ELINT: The Interception and Analysis Of Radar Signalss*. Norwood, MA, USA: Artech House, 2006.

[16] M. Zhu, Y. Li, and S. Wang, "Model-based time series clustering and interpulse modulation parameter estimation of multifunction radar pulse sequences," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 57, no. 6, pp. 3673–3690, Dec. 2021, doi: 10.1109/TAES.2021.3082660.

[17] H. K. Mardia, "New techniques for the deinterleaving of repetitive sequences," *IEE Proc. F Radar Signal Process.*, vol. 136, no. 4, pp. 149–154, Aug. 1989.

[18] D. J. Milojevic and B. M. Popovic, "Improved algorithm for the deinterleaving of radar pulse," *IEE Proc. F Radar Signal Processing*, vol. 139, no. 1, pp. 98–104, 1992.

[19] K. Nishiguchi and M. Kobayashi, "Improved algorithm for estimating pulse repetition intervals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 2, pp. 409–421, Apr. 2000.

[20] J.-W. Tao, C.-Z. Yang, and C.-W. Xu, "Estimation of PRI stagger in case of missing observations," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7982–8001, Nov. 2020.

[21] Y. Liu and Q. Zhang, "Improved method for deinterleaving radar signals and estimating PRI values," *IET Radar, Sonar Navigat.*, vol. 12, no. 5, pp. 506–514, May 2018.

[22] N. Visnevski, V. Krishnamurthy, A. Wang, and S. Haykin, "Syntactic modeling and signal processing of multifunction radars: A stochastic context-free grammar approach," *Proc. IEEE*, vol. 95, no. 5, pp. 1000–1025, May 2007.

[23] N. Visnevski, V. Krishnamurthy, S. Haykin, B. Currie, F. Dilkes, and P. Lavoie, "Multi-function radar emitter modelling: A stochastic discrete event system approach," in *Proc. 42nd IEEE Int. Conf. Decis. Control*, Maui, HI, USA, Dec. 2003, pp. 6295–6300.

[24] L. Zhangmeng, Y. Shuo, and K. Shiqian, "Semantic coding and model reconstruction of multifunctional radar pulse train," *J. Radars*, vol. 10, no. 4, pp. 559–570, 2021.

[25] S. Yuan, S.-Q. Kang, W.-X. Shang, and Z.-M. Liu, "Reconstruction of radar pulse repetition pattern via semantically coding intercepted pulse trains," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Jun. 30, 2022, doi: 10.1109/TAES.2022.3187385.

[26] P. D. Grunwald, *The Minimum Description Length Principle*. Cambridge, MA, USA: MIT Press, 2007.

[27] K. Sayood, *Introduction to Data Compression*, 4th ed. Burlington, CA, USA: Morgan Kaufmann, 2012.

[28] F. Zuyun, *Information Theory—Basic Theory and Application*. Beijing, China: Publishing House of Electronics Industry, 2015.

[29] Z. Liu, S. Kang, and X. Chai, "Automatic pulse repetition pattern reconstruction of conventional radars," *IET Radar, Sonar Navigat.*, vol. 15, no. 5, pp. 500–509, May 2021.

[30] H. Thomas and T. Cormen, *Introduction to Algorithms*, 2nd ed. Cambridge, MA, USA: MIT Press, 2001.

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

SHUO YUAN was born in 1997. He received the B.E. degree in electronic engineering from the National University of Defense Technology, Changsha, China, in 2020, where he is currently pursuing the M.E. degree in communication and information engineering. His research interests include radar signal processing and machine learning.

TAO XU was born in 1998. He received the B.E. degree in electronic engineering from the National University of Defense Technology, Changsha, China, in 2020, where he is currently pursuing the M.E. degree in electronic information. His research interests include radar signal processing and machine learning.

MIN ZHANG received the Ph.D. degree from the National University of Defense Technology (NUDT), Hunan, China, in 2014. He is currently an Associate Professor with NUDT. His main research interests include radar signal processing, passive location, and target tracking.

ZHANGMENG LIU received the Ph.D. degree in statistical signal processing from the National University of Defense Technology (NUDT), Hunan, China, in 2012. He is currently a Professor with NUDT working in the interdiscipline of electronics engineering and computer science, especially electronic data mining, and also with the Tianjin Institute of Advanced Technology, Tianjin, China.