CHAPTER 2

GRAMMATICAL MODELING IN RADAR ES

This chapter first presents the basic functionality of a conventional radar ES system, and the challenge of modern radar ES recognition. Section 2.3 explains how SCFGs can be used to model MFRs. Finally, the fundamental principles involved in learning production probabilities of SCFGs is exposed in Section 2.4.

2.1 Traditional radar ES systems

A radar ES system allows for the passive detection and identification of radar signals for military purpose. As shown in Fig. 1, the basic functionality of current radar ES approaches can be decomposed into three tasks – reception of radar signals, grouping of pulses according to emitter, and recognition of corresponding radar types (Wiley, 1993).

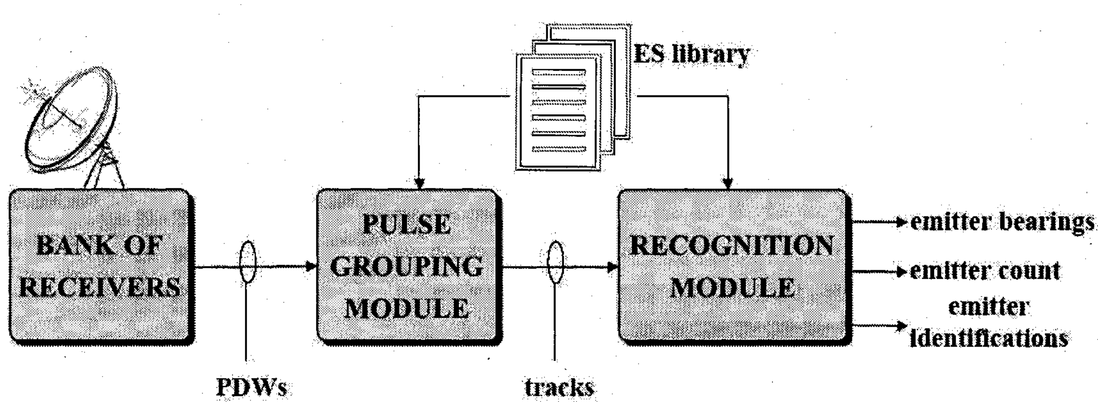


Figure 1 Block diagram of a traditional radar ES system (Granger, 2002).

According to this figure, radar signals are passively intercepted by the receiver portion of the ES system. In typical theaters of operation, intercepted signals are a mixture of electromagnetic pulses transmitted from several emitters. An emitter is an instance of a radar type, and it is not uncommon to observe several emitters of a same type all being active in a theater of operation. A single type of radar can also operate under several different modes to perform various functions. Simultaneous illumination by these emitters causes overlap and interleaving of the received pulses. Upon detection of a radar pulse, most receivers measure the pulse amplitude (PA), pulse width (PW), radio frequency of the carrier wave (RF) and time-of-arrival (TOA). Direction-finding receivers also measure the bearing (Brg), while advanced receivers also measure the modulation on pulse (MOP). Once parameter values have been measured for a pulse, they are digitized and assembled into a data structure called a Pulse Descriptor Word (PDW).

The stream of successive PDWs is fed to a pulse grouping module, which performs either TOA de-interleaving, or sorting, or both. In short, this module seeks to recover pulse trains and their inter-pulse structure prior to further analysis. This involves progressively grouping pulses that appear to have been transmitted from the same emitter. TOA de-interleaving attempts to discover periodicities in the TOA of pulses using techniques such as TOA difference histogramming (Davies and Hollands, 1982; Wiley, 1993). If periodicities are found, and these correlate with radar intelligence compiled in an ES library, then the corresponding pulses are grouped based on PRI, and stripped away from the input stream of PDWs. Sorting attempts to group pulses based on the similarity of their PDW parameters such as RF, PW and Brg. Gating (Davies and Hollands, 1982; Rogers, 1985) or clustering (Anderberg, 1973) techniques are commonly used to this end.

Recognition makes use of an ES library in which are stored the parametric descriptions of known radar types, and attempts to assign a single radar type to each track. Incidentally, the parametric ranges of various types can overlap in the library, and multiple candidates can appear plausible for the same track, a situation known as an “ambiguity." Therefore, a list of likely radar types is often displayed by an operator interface and monitored over time for every track, along with a confidence rating, threat level, latest bearings, and so on. Further analysis can assist an ES operator in revealing mode changes in emitters, links between emitters, and inferred platforms.

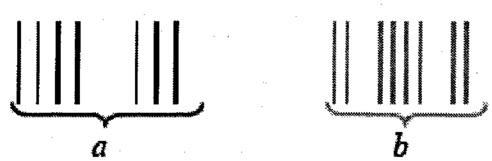
2.2 Challenges of radar ES recognition

Two critical functions of radar ES are the recognition of radar emitters associated with intercepted pulse train, and estimation of the threat level posed by these radars at any given time. The recent proliferation of complex electromagnetic signals encountered in modern environments is greatly complicating these functions. In order to perform these functions, ES systems must keep evolving in response to the agility radar signals, and to power management and low probability of intercept waveforms.

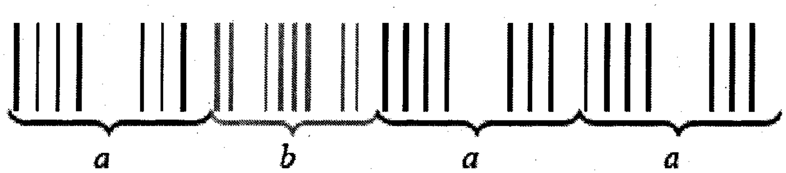
The multiplication of radar modes is the result of computer control and the ease with which parameters such as RF and PRI can be changed. From an ES standpoint, agility in these parameters can make pulse grouping very difficult, and ES libraries very complex. It is difficult and expensive to maintain comprehensive ES libraries that accurately reflect each specific operational environment. Library construction requires explicit modeling of known radar systems, based on prior information and data. This task is complex, tedious, and prone to error. Owing to the multiplication of modes, it is not uncommon for a library to be incomplete and to contain erroneous data. A shorter response time requires faster pulse grouping, as well as recognition using fewer pulses. In addition, the occurrence of low power waveforms implies that pulses near the receiver detection threshold may be dropped, and hence that pulse grouping must work satisfactorily on sparse data. Finally, response time is critical if threats are to be avoided, or self-protection measures such as chaff dispensing, maneuvering, or electronic jamming, are to be successful.

In conventional ES systems, radar signals are often recognized using temporal periodicities within the pulse train in conjunction with histograms of the pulses in some parametric space, e.g., frequency and pulse width. These approaches are ill-suited to exploit the fact that many modern radar systems are highly dynamic and can frequently change their transmitted signals in response to various events. A drawback of histogramming of parameters associated with individual pulses is that most of the temporal relationships amongst the pulses is typically lost. On the other hand, the limitation of periodic temporal analysis is that it assumes that the radar system is a stationary source of pulses. This holds true only for very simple radar systems, and often only over short periods of time.

With the advent of automatic electronic switching designed to optimize radar performance, modern radars, and especially multi-function radars (MFR), are usually far too complex to be recognized using temporal periodicities within the pulse train. MFR will continuously and autonomously switch from one type of signal to another to adapt to the changing environment. Such changes can occur, for example, when the radar detects or abandons targets and consequently switches amongst its search, acquisition and tracking functions, or when a missile is engaged and requires command guidance. The radar emitter is partially driven by the target. Moreover, some electronically steered radar systems may perform many functions simultaneously, greatly increasing the complexity of the radiated signal. Track-While-Scan (TWS) radars and Multi-Function Radars (MFRs) can for instance simultaneously engage multiple targets.



(a) Two MFR words of pulses and identified with symbols a and b.



(b) Sequences of symbols represent concatenated radar words.

Figure 2 Example of the TOA of pulses of two different MFRs.

In light of current challenges in radar ES, more powerful approaches are sought to achieve enhanced accuracy and reliability for radar type recognition, and for instantaneous estimation of threat levels.

2.3 Stochastic grammatical modeling of MFRs

Haykin and Currie (2003) and Lavoie (2001) attempted to apply *Hidden Markov Models* (HMM) to the problem of emitter recognition and threat evaluation. HMMs are a statistical framework for modeling systems that follow a Markov process. It can be defined as a stochastic model in which only observable states are accessible, and whose purpose is to determine the hidden states. In their model, the observable states of the HMM would correspond to time-windowed observations of pulses, while the corresponding radar state (`Search`, `Acquisition`, *etc.*) would represent the hidden states. They concluded that basic HMMs were not suitable for modeling MFR systems when used as the only processor, because of the complexity of the MFRs and their adaptive behaviours to the environment. They added that a hierarchical structure is needed in order to perform the different tasks of an emitter recognition system, consisting in a word recognizer, a sequence recognizer, and a state estimator.

A HMM can be seen as a particular simple type of grammar called *Regular Grammars* (RG). Using a *Context-Free Grammar* (CFG) – a more complex class of grammars – may provide an efficient means to model MFR systems. In particular, signal processing algorithms based on *Stochastic Context-Free Grammars* (SCFGs) constitute one promising approach (Dilkes, 2005a; Visnevski *et al.*, 2003) for future ES systems. The rest of this section provides some background information on modeling of MFR with deterministic and stochastic context-free grammars.

In modern radar ES applications, pulsed radar signals are generated by a MFR in reaction to its current operating environment. For instance, when a radar detects or abandons targets it switches among its `Search`, `Acquisition` and `Tracking` functions – also named radar states. The algorithm controlling the function of a MFR is designed according to stochastic automata principles, and the state transitions within the automata are driven by the stochastic behavior of the targets (Visnevski, 2005). Consequently, MFR have a finite set of behaviors, but the transition sequence among them is unpredictable. The resulting signals from MFRs may be decomposed into two levels of data organization – the *pulse level*, and the *word level*. Radar *words* can be defined as certain static or dynamically-varying groups of pulses that a MFR emits in different states, as shown in Fig. 2(a). In addition, a concatenated sequence of several words may form a *phrase*, which corresponds to a state of the radar. The number of words per phrase, their structure, etc., varies according to the MFR.

A deterministic formal *language* Lg is defined to be a set of finite sequences of symbols drawn from some finite vocabulary V. Linguistic modeling of a radar system's behaviour may be achieved if one identifies symbols of the vocabulary with the words of a specific MFR, as illustrated in Fig. 2(a). By concatenating the corresponding words together, as shown in Fig. 2(b), a language may represent all possible sequences of words that a radar could ever emit, from power-up to shutdown. For electronically agile radar systems, the language can be quite sophisticated and does not have a straightforward description. However, one can create a finite set of grammatical rules to describe a particular language associated with complex radar systems. (Visnevski *et al.*, 2005; Visnevski, 2005).

A *grammar* G is a mathematical construction represented by the quadruplet $G = \{V, N, R, Start\}$. It consists of a vocabulary or terminal alphabet V, a set of nonterminals symbols N, a set of production rules R, and a start symbol $Start$. A production rule has the following aspects: $\Upsilon \rightarrow \Gamma$, where Υ and Γ are elements of $(V \cup N)^*$ – which means that they are combinations of undefined length of elements of V and N – and are called sentential forms. The start symbol $Start$ is an element of N. There is a unique empty string represented by ϵ, which is an element of V^*.

It is possible to classify grammars according to one of the four following families in the Chomsky hierarchy (Fu, 1982), as defined by the form of their production rules (in this description, an upper-case letter is an element of N, a lower-case letter an element of V, and a Greek letter is an element of $(V \cup N)^*$):

a. the regular grammars (RG): $A \to aB$, or $A \to a$;

b. the context-free grammars (CFG): $A \to \lambda$;

c. the context-sensitive grammars (CSG): $\Sigma_1 A \Sigma_1 \to \Sigma_1 \lambda \Sigma_2$, where Σ_1 and $\Sigma_2 \in (V \cup N)^*$ and $\lambda \in (V \cup N)^* \setminus \{\epsilon\}$;

d. the unrestricted grammars (UG): not defined by any specific rule.

Thus, the Chomsky hierarchy can be summarized by: RG $\subset$ CFG $\subset$ CSG $\subset$ UG. Consider a *Context-Free Grammar* (CFG) G, corresponding to the four-tuple $\{V, N, R, \mathit{Start}\}$, where $N = \{\mathit{Start}, A, B, ..., C\}$ is a finite set of non-terminal symbols, $V = \{a, b, ..., c\}$ is a finite set of terminal symbols ($V \cap N = \emptyset$), $\mathit{Start} \in N$ is the initial non-terminal symbol, and R is a finite set of rules of the form $A \to \lambda$ where $A \in N$, $\lambda \in (V \cup N)^*$. Only grammars with no empty rules are considered here.

A *derivation tree* d_x, of a sequence $x \in V^*$ in G, is a sequence of rules $(r_x^1, r_x^2, ..., r_x^m) = d_x$, $m \geq 1$, such that the x is generated from the Start symbol, by successively generating combinations of terminals and non-terminals $\Sigma_i \in (V \cup N)^*$: $(\mathit{Start} \overset{r_x^1}{\Rightarrow} \Sigma_1 \overset{r_x^2}{\Rightarrow} \Sigma_2 \overset{r_x^3}{\Rightarrow} ... \overset{r_x^m}{\Rightarrow} x)$. The *language* generated by G is defined as $Lg(G) = \{x \in V^* | \mathit{Start} \Rightarrow x\}$, that is the set of terminals that can be derived from Start by applying the production rules in R – it can also be seen as a particular subset of V^*. In the example of Fig. 3 (a), $d_x = (r_x^1 \equiv A \to AcA, r_x^2 \equiv A \to AbA, r_x^3 \equiv A \to a, r_x^4 \equiv A \to a, r_x^5 \equiv A \to a)$, that gives $(A \overset{r_x^1}{\Rightarrow} AcA \overset{r_x^2}{\Rightarrow} AbAcA \overset{r_x^3}{\Rightarrow} abAcA \overset{r_x^4}{\Rightarrow} abacA \overset{r_x^5}{\Rightarrow} abaca)$(note that the production rules were applied from left to right on the intermediary sequences of symbols, otherwise the same set of production rules may lead to another sequence of terminal symbols). A CFG is said to be *unambiguous*, if for each $x \in Lg(G)$, there exists only one derivation; otherwise it is called ambiguous.

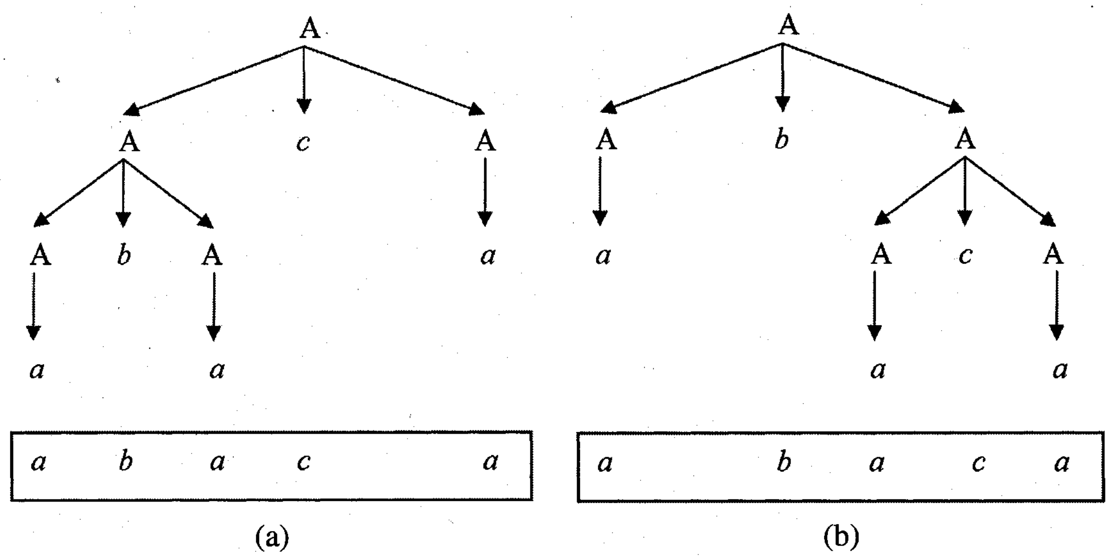


Figure 3 Two derivation tree of the sequence "a b a c a", given the context-free grammar A $\rightarrow$ A b A | A c A | a. Here, A is identified with the *Start* symbol, even if it also appears on the right side of production rules.

For each sequence $x \in Lg(G)$, let $\overline{\Delta}_x$ represent the set of all possible derivation trees that the grammar G admits, starting with $Start$ and leading to x. Hereafter, $\Delta_x \subset \overline{\Delta}_x$ is some selected subset of derivation trees over x.

For each production rule $A \rightarrow \lambda$ in R, and derivation tree d_x, let $N(A \rightarrow \lambda, d_x)$ denote the number of times that $A \rightarrow \lambda$ appears in d_x. Then the total number of times that the non-terminal symbol A appears in d_x is given by:

$$N(A, d_x) = \sum_{\lambda} N(A \rightarrow \lambda, d_x) \tag{2.1}$$

where the sum is extended over all sentential forms λ for which $A \rightarrow \lambda$ appears in R. In the example tree d_x shown in Fig. 3 (a), one has $N(A, d_x) = 5$ and $N(A \rightarrow a, d_x) = 3$.

At a word level, most MFR systems of interest have a natural and compact description in terms of CFGs. Therefore, a CFG allows to model long term dependencies established between the different words of a MFR sequence. However, given the behavior of MFRs and the imperfections of signals observed on a battlefield, it is not possible to design a robust deterministic CFG to model the behavior of a radar system. To robustly model the signal degradations, noise and uncertainties, an element of stochasticity is introduced into the definition of grammars by assigning probability distributions to the production rules. In *Stochastic Context-Free Grammars* (SCFGs) (Fu, 1982) every production for a non-terminal A has an associated probability value such that a probability distribution exists over the set of productions for A. It incorporates stochastic information that allows for a robust modeling of the signal degradations, noise and uncertainties. SCFGs form an important class of grammars which are widely used to characterize the probabilistic modeling of language in computational linguistic and automatic speech recognition and understanding (Fu, 1982), or in RNA secondary structure prediction (Dowell and Eddy, 2004; Sakakibara *et al.*, 1994).

A SCFG G_s is defined as a pair (G, π) where G is a CFG and $\pi = (\pi_{A_1}, \pi_{A_2}, ..., \pi_{A_r})$ is a vector of probabilities whose each element π_{A_i} represents the distribution of probabilities of a nonterminal A_i producing a combination of symbols λ. So $\theta(A_i \rightarrow \lambda)$ is the probability of A_i producing λ and $\pi_{A_i} = (\theta(A_i \rightarrow \lambda), \theta(A_i \rightarrow \mu), ..., \theta(A_i \rightarrow \sigma))$, where $0 \leq \theta(A_i \rightarrow \lambda) \leq 1$ for λ, and $\sum_\lambda \theta(A_i \rightarrow \lambda) = 1$.

The probability of one derivation d_x of the sequence x of terminal symbols is defined as:

$$P(x, d_x|G_s) = \prod_A \prod_\lambda \theta(A \rightarrow \lambda)^{N(A \rightarrow \lambda, d_x)} \tag{2.2}$$

It corresponds to the product of the probability application functions of all the rules used in the derivation d_x. The *probability of the sequence* x with respect to a specified set of possible derivations Δ_x is defined as:

$$P(x, \Delta_x | G_s) = \sum_{d_x \in \Delta_x} P(x, d_x | G_s) \tag{2.3}$$

and the *probability of the best derivation of the sequence* x from the set of all derivations Δ_x is defined as:

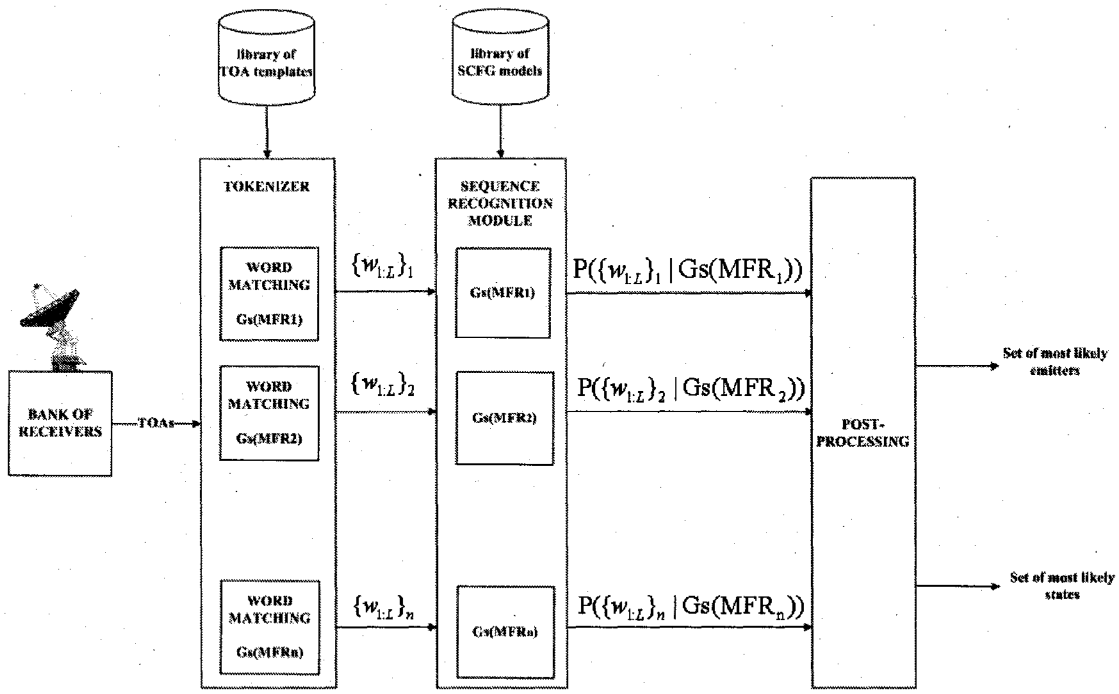
$$\widehat{P}(x | G_s) = \max_{d_x \in \Delta_x} P(x, d_x | G_s) \tag{2.4}$$

Finally, the *best derivation*, $\widehat{d_x}$, is defined as the argument that maximizes Eq. 2.4.

The language $Lg(G_s)$ generated by an SCFG G_s is equal to the language generated by the corresponding CFG G. An important property for any transition probabilities estimation technique is consistency. A SCFG G_s is said to be consistent if $\sum_{x \in Lg(G_s)} P(x|G_s) = 1$ (Sanchez and Benedi, 1997; Fu, 1982).

Fig. 4 shows the block diagram of a radar ES system for recognition of MFRs associated with intercepted pulse trains, and for estimation of the states associated with these MFRs. In this system, a SCFG G_s is used to model each MFR system at a word level only, and therefore would perform the task of sequence recognition and state estimation. In order to perform word recognition, the TOA measured on each incoming pulse sequence is fed to a tokenizer, which performs template matching using, for example, a cross-correlation technique (Dilkes, 2005b; Elton, 2001). Template matching is performed between a window of incoming pulses and the set of words for each MFR. The result is a sequence of words $\{w_{1:L}\}$ for each model of MFR, corresponding to the most likely sequences.

In order to detect the words of a given radar signal, Elton (2001) proposed a cross-correlation (CC) technique. Based on prior information stored in a library, the cross-correlation compares the TOA of pulses in the radar signal with the TOA templates of



(a) ES system.

Figure 4 Block diagram of an ES system that exploits SCFG to model MFR dynamics.

the library. Assuming that the library is composed of the templates that correspond to the words of a radar, the de-interleaving is performed using:

$$R_{sx}(\tau) = \int_{-\infty}^{+\infty} s(t)x(t+\tau)dt \tag{2.5}$$

where $s(t)$ represents the TOA of received pulses and $x(t)$ is TOA template. An example of the signal of pulses produced by this operation for an MFR word is given in Fig. 5. When a sequence of pulses is presented to the tokenizer, the probability of appearance of each MFR word is displayed with respect to the TOA of the pulses. In other words, a peak indicates that a replication of the word begins at the pulse corresponding to this TOA. The input sequence, for this example, corresponds to MFR words that each have a fixed PRI and a fixed number of pulses. Dilkes (2004b) extends the CC technique for noisy radar data.

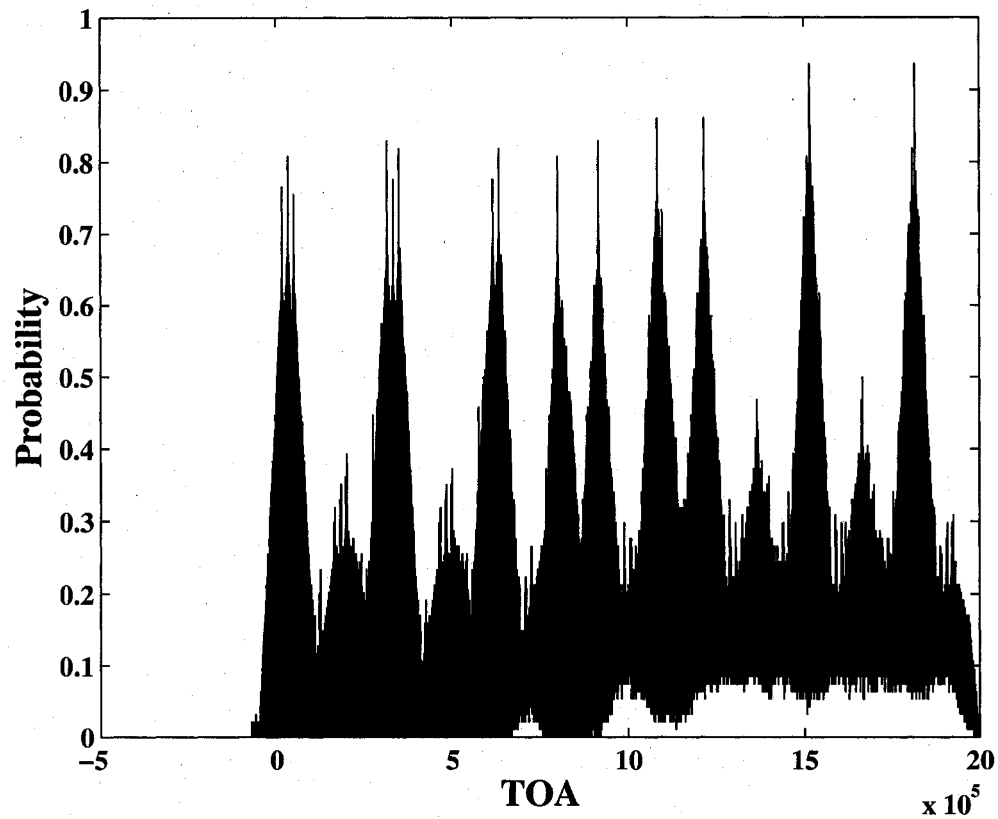


Figure 5 An example of the signal produced for a sequence of pulses that corresponds to a MFR word with fixed PRI, via cross-correlation technique. It represents the probability that the word starts at a TOA, versus the TOA of the pulses.

Once a pattern of pulses is successfully associated with an MFR word, the word replaces the sequence $\{w_{1:L}\}$ corresponding to the MFR. The sequence is then fed to a sequence recognition module. This module computes the probability $P(\{w_{1:L}\}|G_s(MFR))$ that the SCFG G_s associated with each MFR has generated the incoming sequence of MFR words. The sequence recognition module has access to predefined word-level SCFG models, each one corresponding the dynamic behavior of a MFR of interest. If the probability of a SCFG remains above some pre-defined decision threshold for a sequence of words, one can conclude that it corresponds to the radar type associated with the grammar. In addition, the sequence of words can provide an estimate of that radar's state, and therefore its instantaneous level of threat.

2.4 Learning production probabilities of SCFGs

Although SCFGs provide a natural framework for the description of MFR system, the computational and memory requirements of their signal processing algorithms are generally high. One area of concern is the learning of SCFG rules and/or probability distributions associated with the rules, given some prior knowledge and a set of training sequences. The most popular techniques require very high computational time and memory requirements, that make them unsuitable for radar ES applications. Faster techniques have therefore to be investigated.

SCFGs learning techniques could be integrated into a suite of software tools to assist an ES analyst in the construction of grammatical MFR models for a given theater of operation. The choice of a specific technique depends on the level of prior information to construct the SCFGs. If the analyst knows the basic CFG structure for a MFR of interest, he can learn the production rule probabilities based on a set of training sequences collected in the field. Otherwise, he must also learn the grammatical rules for the CFG (although outside the scope of this report, it is worth noting that grammatical inference techniques have been proposed for learning the rules and probabilities of a SCFG (Sakakibara, 1990) (Nakamura and Matsumoto, 2002)). Finally, if a SCFG has previously been designed, an analyst can incrementally learn a new training sequence that becomes available. If new rules are needed for a new sequence, he can simply add them to the grammar. This has no impact on the previous technique since this rule would not have been used.

This thesis is focused on efficient techniques for learning production rule probabilities of a SCFG. Learning production rule probabilities from training sequences is particularly suitable for complex environments, where explicit modeling is difficult. Indeed, the resulting systems can learn and generalize from examples the rules required for MFR recognition. However, their performance depends heavily on the availability of representative training data, and the acquisition of a such a training set is expensive and time consuming in practical applications. Data presented to the ES system in Fig. 4, during either the training or operational phases, may therefore be incomplete in one or more ways.

In ES applications, training data are frequently made available at different points in time. It is therefore highly desirable to update the production rule probabilities of SCFG in an incremental fashion to accommodate the new training data, without compromising the performance. Furthermore, it is not practical in the current setting to accumulate and store all training data in memory, and to retrain a SCFG using all cumulative data. An incremental learning algorithm is the one that meets the following criteria:

a. it should be able to learn additional information from new training sequences;

b. it should not require access to the original training sequences, used to learn the existing SCFG;

c. it should preserve previously-acquired knowledge, i.e., it should not suffer from catastrophic forgetting.

In this thesis, the following approach is considered for designing and maintaining a SCFG to model the dynamics of a MFR:

a. **Initial SCFG design**: define the set of rules of the SCFG to describe the MFR's operation at a word level, and initialize production rule probabilities, either randomly, or based on prior domain knowledge;

b. **Off-line learning**: learn the production rule probabilities parameters of the SCFG based on a representative set of training sequences gathered from the theater of operation;

c. **Incremental learning:** as new training sequences are progressively gathered for the field, perform incremental learning to update and refine existing production rule probabilities of the SCFG based on intercepted sequences from the field. This phase could also involve automatically proposing suitable incremental modifications to the grammatical production rules.