Signal Interpretation of Multifunction Radars: Modeling and Statistical Signal Processing With Stochastic Context Free Grammar

Alex Wang and Vikram Krishnamurthy, *Fellow, IEEE*

Abstract—**Multifunction radars (MFRs) are sophisticated sensors with complex dynamical modes that are widely used in surveillance and tracking. Because of their agility, a new solution to the interpretation of radar signal is critical to aircraft survivability and successful mission completion. The MFRs' three main characteristics that make their signal interpretation challenging are: i) MFRs' behavior is mission dependent, that is, selection of different radar tasks in similar tactic environment given different policies of operation; ii) MFRs' control mechanism is hierarchical and their top level commands often require symbolic representation; and iii) MFRs are event driven and difference and differential equations are often not adequate. Our approach to overcome these challenges is to employ knowledge-based statistical signal processing with syntactic domain knowledge representation: a signal-to-symbol transformer maps raw radar pulses into abstract symbols, and a symbolic inference engine interprets the syntactic structure of the symbols and estimates the state of the MFR. In particular, we model MFRs as systems that "speak" a language that can be characterized by a Markov modulated stochastic context free grammar (SCFG). We demonstrate that SCFG, modulated by a Markov chain, serves as an adequate knowledge representation of MFRs' dynamics. We then deal with the statistical signal interpretation, the threat evaluation, of the MFR signal. Two statistical estimation algorithms for MFR signal are derived—a maximum likelihood sequence estimator to estimate the system state, and a maximum likelihood parameter estimator to infer the system parameter values. Based on the interpreted radar signal, the interaction dynamics between the MFR and the target is studied and the control of the aircraft's maneuvering models is implemented.**

Index Terms—**Electronic warfare, inside-outside algorithm, Galton–Watson branching process, maximum-likelihood estimation, multifunction radar, stochastic context-free grammars, syntactic modeling, syntactic pattern recognition.**

I. Introduction

ELECTRONIC support measure, a division of electronic warfare, involves intercepting and interpreting radiated electromagnetic energy for an operational commander to locate and identify radar sources, and evaluate their potential threats. The electronic support algorithm described in this paper considers the self protection of the target from radar threats, and a major component of which is the interpretation of the intercepted radar pulses in terms of the possible radar modes, such as "search" and "track maintenance." In the current problem setup, because we focus on the target perspective, the radar model is simplified by removing its multiple target tracking capability, and we limit the scenario to having only one multifunction radar in the proximity of the target.

In building electronic support systems to analyze radar signals, statistical pattern recognition has been used extensively. Conventional radars could be characterized by fixed parameters such as radio frequency, pulsewidth, and peak amplitude [1], [2]. For such radar characterizations, decision-theoretic approach as in statistical pattern recognition is sufficient for solving signal processing problems such as emitter identification and threat evaluation. References [3] and [4] discuss template matching of the intercepted radar signal against an EW library for both emitter type and emitter mode identification. Histogram techniques are described in [5] to study temporal periodicities in radar signals such as pulse repetition intervals.

However, modern radars, especially multifunction radars (MFRs), makes the statistical pattern recognition approach inadequate. MFRs are radio-frequency sensors with beam-steering antennas that are widely used in modern surveillance and tracking systems, and they have the capability to perform a multitude of different tasks simultaneously by multiplexing them in time using short time slices [6]. The list of these tasks includes search, acquisition, multiple target tracking, and weapon guidance [7]. At the same time, they maintain low probability of being detected and jammed.

The reasons for the inadequacy of the statistical pattern recognition are two folds. The first concerns with the exploding dimension of the feature space due to the versatility of the radar. The possible variation of the radar parameters such as the carrier frequency and radar pulsewidth makes the statistical pattern recognition infeasible. The second reason deals with the possible time varying feature space necessary for correct recognition. Because of the time multiplexing capability of the radar, the underlying representation of the radar may need to vary in order to capture the dynamics of the radar.

This paper considers a hybrid algorithm of both statistical and syntactical pattern recognition techniques. The methodology is to codify all *a priori* knowledge available and analyze observables within the context of the *a priori* knowledge. Because of the success of formal language in codifying human language, we propose to embody radar domain knowledge in a modified

Manuscript received November 26, 2006; revised July 16, 2007. This work was supported in part by the Department of Defense Canada. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Steven M. Kay.

The authors are with the Department of Electrical Engineering, University of British Columbia, Vancouver, BC V6T 1Z4, Canada (e-mail: alexw@ece.ubc.ca; vikramk@ece.ubc.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2007.908949

language representation, and implement signal interpretation as a *parsing* operation through the radar pulses. In this representation, radar pulses are analogous to English letters, and control rules of pulse generation to English grammar.

The origins of syntactic modeling can be traced to the classic works of Noam Chomsky on formal languages and transformational grammars [8]–[11]. Among the many grammars and languages that have been investigated for practical applications, finite state grammar (FSG) and context free grammar (CFG), as well as their stochastic counterparts, stochastic FSG and stochastic CFG, are currently the most widely used classes of grammars. The application of the grammars to syntactic pattern recognition is covered in depth in [12]. In [13], stochastic context free grammar (SCFG) is applied to study gesture recognition and monitoring of an online parking lot. In [14] and [15], the dynamics of a bursty wireless communications channel is modeled in SCFG. References [16] and [17] describe syntactic modeling applied to bioinformatics and [18] and [19] apply these models to the study of biological sequence analysis and RNA. In addition, on a more related topic to our paper, SCFG is studied in [20] and [21] as an alternative approach to plan recognition.

In this paper, we model MFRs as Markov-modulated SCFGs to take into account the MFR's mode dependent behavior, its hierarchical control, and the control law consisting of operational rules. The more traditional approach such as hidden Markov and state space models are suitable for target modeling [22], [23], but not radar modeling. Traditionally, MFRs' signal modes were represented by volumes of parameterized data records known as electronic intelligence (ELINT) [1]. The data records are annotated by lines of text explaining when, why and how a signal may change from one mode to another. This makes radar mode estimation and threat evaluation fairly difficult. In [24] and [25], SCFG is introduced as a framework to model MFRs' signal and it is shown that MFRs' dynamic behavior can be explicitly described using a finite set of rules corresponding to the production rules of the SCFG. SCFG has several potential advantages that follow.

i) SCFG is a compact formal representation that forms a homogeneous basis for modeling and storing complex system domain knowledge [12], [26], [27], and in which it is simpler and more natural for the model designer to express the control rules of MFR [24]. Specifying the production rules of the SCFG allows convenient modeling of the human computer interface.
ii) SCFG is more efficient in modeling hidden branching processes when compared to a stochastic regular grammars or hidden Markov models with the same number of parameters. The predictive power of a SCFG measured in terms of entropy is greater than that of the stochastic regular grammar [28]. SCFG is equivalent to a multitype Galton-Watson branching process with finite number of rewrite rules, and its entropy calculation is discussed in [29].
iii) The recursive embedding structure of MFRs' control rules is more naturally modeled in SCFG. As we will show later, the Markovian type model has dependency that has variable length, and the growing state space is difficult to handle since the maximum range dependency must be considered.

In summary, the main results of the paper are as follows.

1) A careful detailed model of the dynamics of an MFR using formal language production rules. By modeling the MFR dynamics using a linguistic formalism such as a SCFG, a MFR can be viewed as a discrete event system that "speaks" some known, or partially known, formal language [30]. Observations of radar emissions can be viewed as strings from this language, corrupted by the noise in the observation environment.
2) Novel use of Markov modulated SCFGs to model radar emissions generated by MFR. The complex embedding structure of the radar signal is captured by the linguistic model, SCFG, and the MFR's internal state is modeled by a Markov chain. This modeling approach enables the combination of the grammar's syntactic modeling power with the rich theory of Markov decision process.
3) Statistical signal processing of SCFGs. The threat evaluation problem is reduced to a state estimation problem. Maximum likelihood estimator is derived based on a hybrid of the forward-backward and the inside-outside algorithm. (Inside-outside algorithm is an extension of HMM's forward-backward algorithm [31].)
4) Parameterizing the MFR model with the target's maneuvering models, the interaction between the target and the MFR is studied. The target's probing of the MFR in order to find a maneuvering model that maximize its safety is formulated as a discrete stochastic approximation problem, and simulation study of the problem is performed.

The rest of the paper is organized as follows. Section II describes the multifunction radar in detail and its role in electronic warfare. Section III models the MFR's command generation mechanism, where the construction of the Markov chain in terms of the MFR's goals and subgoals, and MFRs' hierarchical control as a set of syntactic rules are detailed. Section IV presents the threat estimation algorithm and the discrete stochastic approximation algorithm, and Section V provides the numerical studies. Finally, Section VI concludes the paper.

II. Electronic Support and MFR

Electronic warfare (EW) can be broadly defined as any military action with the objective of controlling the electromagnetic spectrum [32]. An important aspect of EW is the radar-target interaction. In general, this interaction can be examined from two entirely different viewpoints, that of the radar and of the target. From the radar's viewpoint, the goal is to detect and identify targets, and to maintain a firm track. From the target's viewpoint, the goal is to protect itself from radar-equipped threat by interpreting intercepted radar emissions and evaluating their threat (electronic support or ES). In this paper, the target's viewpoint is the focus, and MFRs are the specific threat considered.

The approach taken in this paper to interpret the MFR signal is knowledge-based. The raw radar signal is interpreted with respect to a grammatical model that describes its characteristics; the characteristics of interest is the order of the events detected, and the event occurrence time is not of much importance. The signal interpretation consists of two main components, a signal-to-symbol transformer and a symbolic inference engine. Fig. 1 illustrates the two components in the context of the ES architecture, and a brief description of which is given here: The

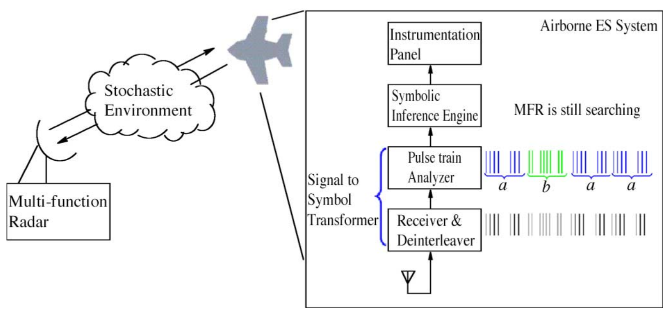


Fig. 1. Electronic support (ES) framework considered in this paper. The radar signal emitted by the MFR is captured by the ES system on board the target after being corrupted by the stochastic environment. The system consists of an antenna, a signal-to-symbol transformer and a symbolic inference engine. The signal-to-symbol transformer consists of a receiver/deinterleaver and a pulse train analyzer, and its main purpose is to map raw radar signal to abstract symbols that are recognizable by the symbolic inference engine. The symbols are identified as a and b in the figure.

receiver processes the radar pulses intercepted by the antenna, and outputs a sequence of pulse descriptor words (PDW), where a PDW is a data structure containing parameters such as carrier frequency, pulse amplitude, and pulsewidth of an individual pulse. The PDWs are then processed by the deinterleaver, and segregated according to their originating radar emitters. The pulse train analyzer further processed the deinterleaved PDWs, and classify them into abstract symbols called radar words. (See Section II-A for definitions.) Finally, the symbolic inference engine analyzes the syntactic structure between the radar words, interprets its threat level, and outputs the results on a pilot instrumentation panel.

Because the receiver, deinterleaver and pulse train analyzer have been well studied, the signal-to-symbol transformer is not covered in this paper, and we only focus on the symbolic inference engine. Using an analogy between the structural description of the radar signal and the syntax of a human language, a symbolic inference engine is said to contain the prior domain-specific knowledge of the "language" MFRs "speak." The knowledge consists of the operational rules and constraints captured by the radar analysts that are believed to be applied in the generation of the radar signal for each specific mission goal, and such knowledge allows the radar analysts to distinguish "grammatical" radar signal from "ungrammatical" one, and to reason about the particular mission goal the MFR is executing. In today's modern radar systems, the operational rules are often implemented with fuzzy logic or expert system [22], and conventional mathematical formalisms such as differential and difference equations are not effective in analyzing them. Instead, in order to compactly store the syntactic knowledge of the MFR's language, formal language theory is applied, and the MFR language would be fully specified by the establishment of a grammar [27].

As far as ES is concerned, the optimal approach is to collect a corpus of radar samples, and induce the grammar directly without human intervention. However, because of the degree of complexity and potentially lack of data on the MFR signal, grammatical induction approach is impractical. In this paper, stochastic context free grammar is chosen to model the MFR signal for each of its mission goal because of its generality over the hidden Markov and state space models, and the existence

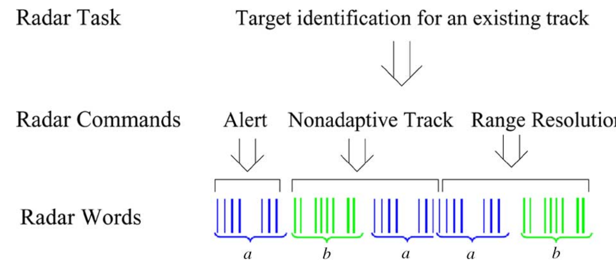


Fig. 2. Radar signal corresponds to different layers of radar command generation hierarchy. A radar task consists of a sequence of radar commands that would best achieve a tactic goal, and each radar command can be mapped to a certain catenation of radar words that MFR is to execute.

of algorithms for parameter estimation. The context-free backbone is constructed from the domain-specific knowledge of the MFRs' signal generation mechanism. Section II-A describes the MFR's domain-specific knowledge that would be used to construct the model for knowledge-based signal processing.

A. MFR System Architecture and Its Signal Generation Mechanism

Before discussing the MFR architecture, we begin by describing the radar signal that is generated by different layers of the MFR command generation hierarchy. The list below begins by the actual radar pulses generated by the MFR, to the software objects that are scheduled by the MFR processor, and ends with the radar policy that governs the scheduling process.

- *Radar word*: A fixed arrangement of finite number of pulses. For example pulses with a fixed pulse repetition frequency.
- *Radar command*: Catenation of finite number of radar words that is optimized for extracting certain target information. Examples are target acquisition and nonadaptive track.
- *Radar task*: The three main radar tasks are search, target identification and target tracking, and each is implemented by a template of radar commands designed to achieve the tactical goal.
- *Radar mode*: The constraints or emphasis on the execution of certain radar tasks due to the mission requirements or resource allocations.

An example of the above radar signal is illustrated in Fig. 2. The radar task and the radar commands in the example are self-explanatory, and the letters a and b denote radar words. The vertical bars represent radar pulses, and a particular arrangement of them makes up the radar words.

Following the macro/micro architecture as described in [22, Section 15.5.6 15.5.6], the generation of the radar signal is modeled by a MFR composed of four basic components:[1] a situation assessment, a radar manager, a command scheduler, and a radar controller, which are illustrated in Fig. 3. The chain of commands starts with the situation assessment which provides evaluation of the tactic environment to the radar manager. The radar manager evaluates the threat accordingly, and enters the appropriate radar task to the planning queue for scheduling. The radar

[1]The system architecture does not include multiple target tracking functionalities such as data association. The paper focuses on a single target's self protection and threat estimation, and thus models only the radar signal that a single target can observe.

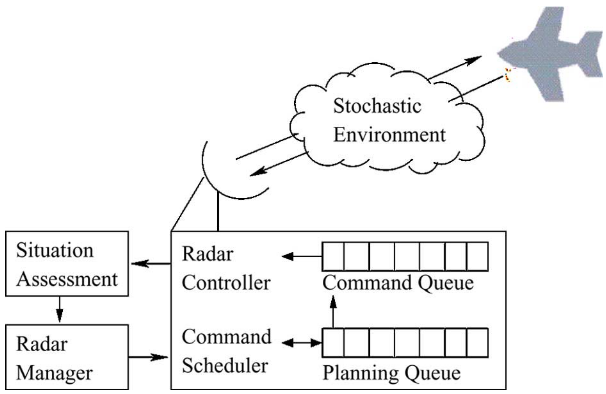


Fig. 3. MFR system architecture. The situation assessment provides the evaluation of tactical environment to the radar manager. The radar manager, based on the evaluation, selects a radar task on which the command scheduler/radar controller will operate. The command scheduler plans and preempts the tasks in the planning queue depending on the radar load, and the moves the tasks fixed for execution to the command queue. The radar controller maps the tasks in the command queue to appropriate radar commands, which is retrieved by the radar for final execution.

task consists of a sequence of macro radar commands, and the commands can be repeated or preempted in the planning queue by the command scheduler. The commands that are fixed for execution are passed to the radar controller, where they will be mapped to the appropriate radar words and retrieved by the radar for execution.

In the rest of the section, we will discuss the operational details of each of the MFR components, and their relationship to the macro/micro architecture. More specifically, the macro sensor management, as described in [22], requires the MFR to have three basic components: an operating scheme, a performance standard, and an adaptation procedure, and the micro sensor management requires the MFR to be able to select combination of radar pulses that best accomplish the performance requested by the macro tasks given the system status. We will describe how each of the requirements are satisfied by the MFR components.

The macro management is accomplished by the radar manager and the command scheduler. Radar manager sets the operating scheme and the performance standard for the MFR. It is a finite state machine that transitions among a set of tasks, with the transition probabilities determined by the radar mode. It sets the guidance to which radar commands are to be created by mapping each radar task to a template of radar commands. The mapping can be mission dependent, and such dependency models the performance standard. For example, a radar task "Target identification for an existing track," depending on the performance standard, may be mapped to an template of radar commands such as {Alert, Nonadaptive track, Range resolution 1} or {Alert, Nonadaptive track, Range resolution 2}, where Range resolution 1 and 2 differ in carrier frequency and the radar waveforms used.

The command scheduler models the adaptation procedure, and the adaptation is modeled by the scheduler's ability to plan and preempt radar commands in the planning queue. The command scheduler processes the radar commands stored in the planning queue sequentially, and it plans, if the current command requests it, by appending radar commands in the planning queue, and preempts by inserting commands in front of the current command. The planning and preempting will be discussed according to some rules to be specified.

The micro sensor management, on the other hand, is accomplished by the radar controller. Similar to the command scheduler, the radar controller processes the radar commands in the command queue sequentially and maps the radar commands to radar words according to a set of control rules. Each radar command may be mapped to a multitude of different radar words depending on the tactic environment, and the mapping will be specified explicitly later in terms of the grammar's productions in Section III.

As a remark, the control is separated into the command scheduler and the radar controller because of the MFR needs to be both adaptive and fast [33]. The command scheduler orders radar commands by time and priority, and stores them in the planning queue for it allows real time rescheduling. On the other hand, due to the system's finite response time, radar commands in the planning queue are retrieved sequentially and placed in the command queue where no further planning or adaptation is allowed. The radar controller maps the radar commands in the command queue to radar words and which are retrieved by the radar for execution.

III. A Syntactic Representation of MFR Domain Knowledge

In terms of natural language processing, we model the MFR as a system that "speaks" according to a stochastic grammar, and more specifically, we place the domain knowledge discussed in the previous section in a compact mathematical formalism called the stochastic context free grammar. In Section III-A, an overview of the formal language theory is provided. In Section III-B, the radar manager, the command scheduler and the radar controller are modeled, and the details of the Markov modulated SCFG are provided. In Section III-C, a well posedness issue of the grammatical model is discussed.

A. Formal Languages and Transformational Grammars

A formal language can be broadly defined as any set of strings consisting of concatenations of symbols. The complete set of distinguishable symbols in the language is known as the alphabet and is denoted here by T. For example, an alphabet might be $T = \{a, b\}$, and one language over this alphabet might consist of all finite (or null) repetitions of the combinations ab followed by either b or aa; in this language, the strings b, aa, $ababaa$ and $ababb$ are valid strings but aba is not.

The general notion of a formal language is impractically broad. It is much more useful, and intuitive, to specify a language in terms of its structural patterns. This is often accomplished by defining a grammar [8], [10], [11] sometimes known in the literature as a transformational grammar. In grammatical terminology, a grammar is a four-tuple $\langle N, T, P, S\rangle$. N is a finite set of nonterminal symbols, T is a finite set of terminal symbols, and $N \cap T = \emptyset$. P is a finite set of production rules, and $S \in N$ is the starting symbol. The grammars are divided into four different types according to the forms of their production rules [8], [34]. Specifically, context free grammar has production rules P of the form $A \rightarrow \eta$ where $A \in N$ and $\eta \in (N \cup T)^+$; the superscript Σ^+ indicates the set of all finite length strings of symbols in a finite set of symbols Σ, excluding the string of length 0. The rule $A \rightarrow \eta$ indicates the replacement of the nonterminal A by η. In addition, as shown in [10], any context free grammar may be reduced to Chomsky Normal form, and which has production rules of the form $A_i \rightarrow A_j A_k$ and $A_i \rightarrow w$, where A_i, A_j, $A_k \in N$, and $w \in T$. An example of context free grammar in the Chomsky Normal form consists of the following elements:

$$T = \{a, b\} \quad N = \{A_0, A_1\} \quad S = \{A_0\}$$
$$P = \{A_0 \rightarrow A_0 A_1 | b, A_1 \rightarrow a\}$$

where the bar | separates the two production rules, meaning that the nonterminal A_0 may be mapped to either A_0 A_1 or b. Starting from the nonterminal A_0, the strings can be derived by applying production rules to iteratively replace nonterminal symbols with substrings. The preceding example admits the following derivations:

$A_0 \Rightarrow b$
$A_0 \Rightarrow A_0 A_1 \Rightarrow b A_1 \Rightarrow ba$
etc.

As a shorthand notation, the multiple derivation steps in the last derivation above may also be expressed as $S_0 \overset{*}{\Rightarrow} ba$. Furthermore, please note that the notation $\rightarrow$ is used to express production rules, and $\Rightarrow$ is used to represent derivation or replacement of nonterminals in a string.

In addition, as is often the case, a certain amount of uncertainty exists in the process under study. In order to make the model more robust, and also to capture the random effect in the model, probabilities are added to the set of production rules P. Stochastic context free grammar is a four-tuple $\langle N, T, P^s, S \rangle$ with all elements identical to the context free grammar except P^s is a finite set of stochastic production rules. Let A be a nonterminal in N, the probability of its production rule $A \rightarrow \eta$ in P^s is denoted as $P(A \rightarrow \eta)$, and the probabilities must satisfy

$$\sum_{\eta \in \Theta} P(A \rightarrow \eta) = 1$$

where Θ is the set of all right hand sides for A in P^s. For example, the grammar given above may be converted into a stochastic one by assigning the following probabilities to the production rules

$$A_0 \xrightarrow{0.8} A_0 A_1 \quad A_0 \xrightarrow{0.2} b \quad A_1 \xrightarrow{0.1} a A_1 \quad A_1 \xrightarrow{0.9} a.$$

A Simple Example of MFR and Inadequacy of HMM: As compared to conventional radars, MFRs are distinguished by their ability to switch between radar tasks, and plan ahead their courses of actions [33]. As an illustrative example showing the correspondence between the grammar and the MFR, consider production rules of the form: i) $B \rightarrow bB$ and ii) $B \rightarrow AB|BC$, where A, B and C are nonterminals representing radar commands in the planning queue and b is a radar command in the command queue. The rule $B \rightarrow bB$ is interpreted as directing the command scheduler to append b to the command queue, and B in the planning queue. Similarly, $B \rightarrow AB$ is interpreted as delaying the execution of B in the planning queue and insert A in front. Suppose the planning queue contains the radar command B, a possible generation of the radar words is illustrated in Fig. 4. (The figure also illustrates the mapping of the radar commands to some radar words by the radar controller.) It can be seen that as long as the command queue commands appear only to the left of planning queue commands in the rule, the command queue and the planning queue are well represented.

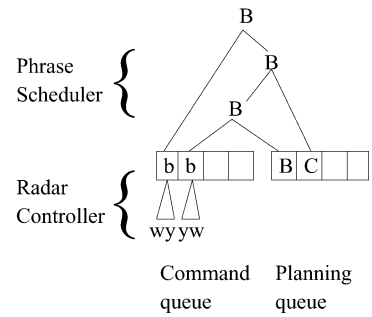


Fig. 4. The figure illustrates a possible realization of the scheduling process represented by a grammatical derivation process. B and C are nonterminals and b is a terminal. The triangle represents the mapping of the radar command b to the radar words, y and w, by the radar controller.

In addition to the interpretation of the production rules, another important property is their generative power, and why a more established method such as hidden Markov model cannot be used. As shown in [35], the rules of the form i have the syntax of regular grammar and they can be used to represent hidden Markov models, i.e., stochastic regular grammar. The rules of the form ii, on the other hand, have the syntax of context free grammar. In other words, the MFR grammar has rules that strictly contain regular grammar (rules of the form ii cannot be reduced to i), and, thus, the MFR grammar cannot be sufficiently modeled by HMM. The production rules presented in this example is a self-embedding context free grammar and it cannot be represented by a Markov chain [10]. A context-free grammar is self-embedding if there exists a nonterminal A such that $A \overset{*}{\Rightarrow} \eta A \beta$ with $\eta, \beta \in (N \cup T)^+$. For the rules presented, self-embedding property can be shown by a simple derivation

$$B \rightarrow AB \rightarrow ABC.$$

In addition to the self-embedding property, HMM is not suitable because the radar controller may generate variable length radar words. If HMM is to model the radar words, the Markovian dependency may be of variable length. In this case, maximum length dependency needs to be used to define the state space, and the exponential growing state space might be an issue. Furthermore, for sources with hidden branching processes (MFRs), stochastic context free grammar is shown to be more efficient than HMM in the sense that the estimated SCFG has lower entropies [28].

B. A Syntactic Model for a MFR Called Mercury

In this subsection, because the MFR domain knowledge is application dependent, for illustrative purpose, the grammatical representation is discussed based on a particular type of MFR called Mercury (The declassified version of the Mercury's textual intelligence report can be found in [36]). The output of the MFR is modeled by a set of terminals, and the hierarchical command generation mechanism is modeled by a set of production rules that map the top level radar tasks to radar commands, and from radar commands to radar words.

TABLE I
LIST OF MERCURY RADAR COMMANDS AND THEIR CORRESPONDING RADAR WORDS

Command	Words	Command	Words
Four-word search	$[w_1w_2w_4w_5]$	Track maintenance (TM)	$[w_1w_7w_7w_7]$
	$[w_2w_4w_5w_1]$		$[w_2w_7w_7w_7]$
	$[w_4w_5w_1w_2]$		$[w_3w_7w_7w_7]$
	$[w_5w_1w_2w_4]$		$[w_4w_7w_7w_7]$
Three-word search	$[w_1w_3w_5w_1]$		$[w_5w_7w_7w_7]$
	$[w_3w_5w_1w_3]$		$[w_6w_7w_7w_7]$
	$[w_5w_1w_3w_5]$		$[w_1w_8w_8w_8]$
Acquisition (ACQ)	$[w_1w_1w_1w_1]$		$[w_2w_8w_8w_8]$
	$[w_2w_2w_2w_2]$		$[w_3w_8w_8w_8]$
	$[w_3w_3w_3w_3]$		$[w_4w_8w_8w_8]$
	$[w_4w_4w_4w_4]$		$[w_5w_8w_8w_8]$
	$[w_5w_5w_5w_5]$		$[w_6w_8w_8w_8]$
Non-adaptive Track (NAT) or Track maintenance (TM)	$[w_1w_6w_6w_6]$		$[w_1w_9w_9w_9]$
	$[w_2w_6w_6w_6]$		$[w_2w_9w_9w_9]$
	$[w_3w_6w_6w_6]$		$[w_3w_9w_9w_9]$
	$[w_4w_6w_6w_6]$		$[w_4w_9w_9w_9]$
	$[w_5w_6w_6w_6]$		$[w_5w_9w_9w_9]$
Range resolution	$[w_7w_6w_6w_6]$		$[w_6w_9w_9w_9]$
	$[w_8w_6w_6w_6]$	Fine track maintenance (FTM)	$[w_7w_7w_7w_7]$
	$[w_9w_6w_6w_6]$		$[w_8w_8w_8w_8]$
ACQ, NAT or FTM	$[w_6w_6w_6w_6]$		$[w_9w_9w_9w_9]$

The MFR grammar is $\{N_r \cup N_p \cup N_c, T_c, P_p \cup P_c, S\}$. N_r is the set of radar tasks. N_p and N_c are identical sets of radar commands available to the MFR, and they are differentiated only by their residing queues; N_p are the commands in the planning queue and N_c are in the command queue. P_p is the set of production rules mapping N_p to $(N_c \cup N_p)^+$. P_c is the set of production rules mapping N_c to T_c^+, where T_c is the set of radar words. In SCFG, S is the starting symbol, however, in our formulation, S is a Markov chain with state space defined by N_r. The output of the Markov chain is in N_p^+ and it is the starting symbols for P_p. Specific to Mercury, the set of radar words T_c consists of nine distinct elements $\{w_1, \ldots, w_9\}$. The set of available radar commands is {Three-word search, Four-word search, Acquisition, Nonadaptive track, three stages of Range resolution, Track maintenance, Fine track maintenance}, and it is written in shorthand as $\{3\mathrm{WS}_t, 4\mathrm{WS}_t, \mathrm{A}_t, \mathrm{NAT}_t, \mathrm{RR1}_t, \mathrm{RR2}_t, \mathrm{RR3}_t, \mathrm{TM}_t, \mathrm{FTM}_t\}$, where t = p or c denoting N_p or N_c respectively. Table I lists the radar commands and their corresponding radar words.

The Mercury's grammar will be introduced according to the framework depicted in Fig. 3. The radar manager is modeled as a Markov chain whose state space is N_r, the command scheduler is represented by the production rule P_p (self-embedding), and the radar controller, introduced along with the effects of the stochastic channel, is modeled by the production rule P_c. We will describe each MFR component in detail.

1) Radar Manager: The radar manager, for each time period, determines the overall task or tactical goal the MFR is to accomplish. The time evolution of the radar manager is modeled as a Markov chain, and its state space, $N_r = \{\text{Search for new targets, Target identificationfor existing tracks, Track update for existing tracks}\}$, is defined based on the major radar task categories [22]. Let $k = 0, 1, \ldots$ denote discrete time. The state of the MFR, $x_k \in N_r$, is a three state discrete time Markov chain. The output of each state is defined by templates of radar commands that specify the type and the order of the radar commands the MFR is to complete in order to accomplish the tactical goal. The templates for the states are expressed in the production rules listed here.

Search for new targets $\rightarrow 3\mathrm{WS}_p|4\mathrm{WS}_p$;
Target identification for existing tracks $\rightarrow \mathrm{A}_p\mathrm{NAT}_p\mathrm{RR1}_p$;
Track update for existing tracks $\rightarrow \mathrm{TM}_p$.

Each state may output multiple templates and they are separated by bars. Different templates are characterized by their computational cost and accuracy, and their selection is modeled probabilistically.

TABLE II
LIST OF TARGET'S MOTION MODELS

Type of Motion Models	s^1	s^2
Constant velocity model	0	0
Time correlated acceleration model	1	0
Horizontal turn model	0	1

Define the transition probability matrix as $A = [a_{ji}]_{3\times 3}$, where $a_{ji} = P(x_k = e_i | x_{k-1} = e_j)$, and e_i and e_j are MFR states in N_r. The transition of the MFR is assumed to be driven by the interaction between the MFR and targets. For example, if the target is far away from the MFR and flies with constant velocity, the probability of the MFR jumping to "Track update for existing tracks" might be low. On the other hand, when the target is close and shows high maneuverability, the probability of being tracked might be higher because MFR would allocate more resources to it.

In order to characterize the interaction between the MFR and a target, the target behavior pattern is described first. A target state process is $\psi_k = (z_k, s_k)$, where z_k refers to its kinematics and s_k is a staircase-type trajectory indicating its motion models such as constant velocity model [37]. In this paper, $z_k \in \mathcal{R}$ denotes distance of the target with respect to the MFR, and $s_k = (s_k^1, s_k^2)$ is an indicator vector featuring the motion model in which the target is maneuvering. The dependency between the MFR and targets is established by parameterizing the transition matrix A with (z_k, s_k).

Table II lists the values of s_k and their corresponding motion models. The list of representative motion models are used in [38] to study the benchmark tracking problem. The first model, constant velocity model, characterizes the periods of nonmaneuverability, and it is described in [39]. The other two models are to account for target maneuvers. The time correlated acceleration model is first proposed in [40] and the horizontal turn model is described in [41].

Because of its generality and utility interpretation, Logit model is selected to parameterize the transition matrix. Let P_{up} (P_{down}) be the probability of the MFR system to move up (down) a state and P_{stay} is the probability of the MFR system remaining in the current state. The probabilities are illustrated in Fig. 5 and they are shown as follows:

$$P_{\mathrm{up}} = \frac{\exp(a'\psi_k)}{1 + \exp(a'\psi_k) + \exp(b'\psi_k)}$$

$$P_{\mathrm{down}} = \frac{\exp(b'\psi_k)}{1 + \exp(a'\psi_k) + \exp(b'\psi_k)}$$

$$P_{\mathrm{stay}} = \frac{1}{1 + \exp(a'\psi_k) + \exp(b'\psi_k)}$$

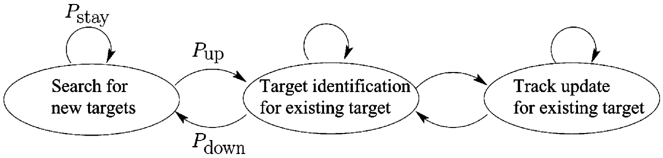


Fig. 5. MFR states and transition probabilities.

TABLE III
PRODUCTION RULES OF MERCURY'S COMMAND SCHEDULER

$3WS_p \rightarrow$	$3WS_c\ 3WS_p \| 3WS_c$
$4WS_p \rightarrow$	$4WS_c\ 4WS_p \| 4WS_c$
$A_p \rightarrow$	$A_c\ A_p \| A_c$
$NAT_p \rightarrow$	$NAT_c\ NAT_p \| NAT_c$
$RR1_p \rightarrow$	$RR1_c\ RR1_p \| RR1_c\ RR2_p \| RR1_c$
$RR2_p \rightarrow$	$RR2_c\ RR2_p \| RR2_c\ RR1_p \| RR2_c\ RR3_p \| RR2_c$
$RR3_p \rightarrow$	$RR3_c\ RR3_p \| RR3_c\ RR2_p \| RR2_p\ RR3_p \| RR3_c$
$TM_p \rightarrow$	$TM_c\ TM_p \| TM_c\ FTM_p \| TM_p\ FTM_p \| FTM_p\ TM_p \| TM_c$
$FTM_p \rightarrow$	$FTM_c\ FTM_p \| FTM_c\ TM_p \| TM_p\ FTM_p \| FTM_c$

where a, and b are vectors of regressor parameters. The justification of the logit model is given in Appendix A.

2) Command Scheduler: The command scheduler models the MFR's ability to plan and to preempt radar commands based on the radar task and the dynamic tactic environment. With the template of radar commands in place, the main operation of the command scheduler is to implement the scheduling of radar commands in the command queue and/or the rescheduling of commands in the planning queue. The operational rules for the scheduling and rescheduling could be constructed based on a small set of basic rules. Suppose $N_p = \{A, B, C\}$ and $N_c = \{a, b, c\}$, the basic control rules that are available to the command scheduler are listed.

Markov and $B \rightarrow bB|bC$
Adaptive and $B \rightarrow AB|BC$
Terminating and $B \rightarrow b$

The interpretation of the rules follows the example given at the end of the previous subsection. A rule is *Markov* if it sent a radar command to the command queue, and re-scheduled either a same or a different radar command in the planning queue. A rule is *Adaptive* if it either preempted a radar command for another radar command or if it scheduled a radar command ahead of time in the radar's time line after the current command. A rule is *Terminating* if it sent a radar command to the command queue without scheduling any new commands.

The significance of the Markov rule is obvious. It represents the completion of one radar command and the scheduling of another. The two adaptive rules model the MFRs' ability to: i) Preempt and ii) Plan the radar commands. The preempt rule is $B \rightarrow AB$, where the command B is preempted when a higher priority task A enters the queue. On the other hand, the plan rule is $B \rightarrow BC$, where the command C is scheduled ahead of time. The terminating rule reflects the fact that the queues have finite length, and the grammatical derivation process must terminate and yield a terminal string of finite length. Applying the basic control rules to the templates, the production rule P_p could be constructed. With some constraints in place, the complete set of rules is listed in Table III.

3) Radar Controller and the Stochastic Channel: The radar command is mapped to the radar words by the radar controller, and the words could be corrupted by the stochastic channel before it's intercepted. Here, production rules of the radar controller are devised, and the effect of the stochastic channel is incorporated.

TABLE IV
PRODUCTION RULES OF MERCURY'S RADAR CONTROLLER

$4WS \rightarrow$	$W_1W_2W_4W_5 \| W_2W_4W_5W_1$ $\| W_4W_5W_1W_2 \| W_5W_1W_2W_4$
$3WS \rightarrow$	$W_1W_3W_5W_1 \| W_3W_5W_1W_3$ $\| W_5W_1W_3W_5$
$A \rightarrow$	$Q_1 \| Q_2 \| Q_3 \| Q_4 \| Q_5 \| Q_6$
$NAT \rightarrow$	$S_1T_6 \| Q_6$
$RR1 \rightarrow$	W_7T_6
$RR2 \rightarrow$	W_8T_6
$RR3 \rightarrow$	W_9T_6
$TM \rightarrow$	$S_1T_6 \| S_2T_7 \| S_2T_8 \| S_2T_9$
$FTM \rightarrow$	$Q_6 \| Q_7 \| Q_8 \| Q_9$
$S_2 \rightarrow$	$S_1 \| W_6$
$S_1 \rightarrow$	$W_1 \| W_2 \| W_3 \| W_4 \| W_5$
$T_6 \rightarrow$	$W_6W_6W_6$
$T_8 \rightarrow$	$W_8W_8W_8$
$T_7 \rightarrow$	$W_7W_7W_7$
$T_9 \rightarrow$	$W_9W_9W_9$
$Q_i \rightarrow$	$W_iW_iW_iW_i$
$W_i \xrightarrow{p_i}$	$w_1 \| w_2 \| w_3 \| w_4 \| w_5 \| w_6 \| w_7 \| w_8 \| w_9$ for $i = 1, \ldots, 9$

The production rules of the radar controller are derived from visual inspection of the radar commands listed in Table I. The syntactic structure of the radar commands are captured by defining the nonterminals and their corresponding production rules. We begin by defining the triplets as follows:

$$T_6 \rightarrow w_6w_6w_6 \quad T_8 \rightarrow w_8w_8w_8$$
$$T_7 \rightarrow w_7w_7w_7 \quad T_9 \rightarrow w_9w_9w_9$$

and blocks of four words

$$Q_1 \rightarrow w_1w_1w_1w_1 \quad Q_4 \rightarrow w_4w_4w_4w_4 \quad Q_7 \rightarrow w_7w_7w_7w_7$$
$$Q_2 \rightarrow w_2w_2w_2w_2 \quad Q_5 \rightarrow w_5w_5w_5w_5 \quad Q_8 \rightarrow w_8w_8w_8w_8$$
$$Q_3 \rightarrow w_3w_3w_3w_3 \quad Q_6 \rightarrow w_6w_6w_6w_6 \quad Q_9 \rightarrow w_9w_9w_9w_9.$$

Furthermore, we introduce two new nonterminals

$$S_1 \rightarrow w_1|w_2|w_3|w_4|w_5 \qquad S_2 \rightarrow S_1|w_6.$$

The nonterminals introduced specifies the complete set of the production rules for the radar controller.

Based on the radar controller's production rules, the effects of the stochastic channel could be easily incorporated. For each radar word w_i, define a new nonterminal W_i and the production rule

$$W_i \rightarrow w_1|w_2|w_3|w_4|w_5|w_6|w_7|w_8|w_9 \ [P_i] \ \text{for } i = 1, \ldots, 9$$

where $P_i = [P_{i1}, P_{i2}, P_{i3}, P_{i4}, P_{i5}, P_{i6}, P_{i7}, P_{i8}, P_{i9}]^T$ is a vector of probabilities indicating how likely W_i would be corrupted and intercepted as one of the other radar words. When compiled together, the complete set of production rules are specified and they are listed in Table IV. As will be illustrated in later sections, the probabilities of the production rules could be estimated based on training data. In addition, since each w_i is a pulse train, a pulse train analysis can be conducted to assign prior probabilities to the channel probabilities W_i [42].

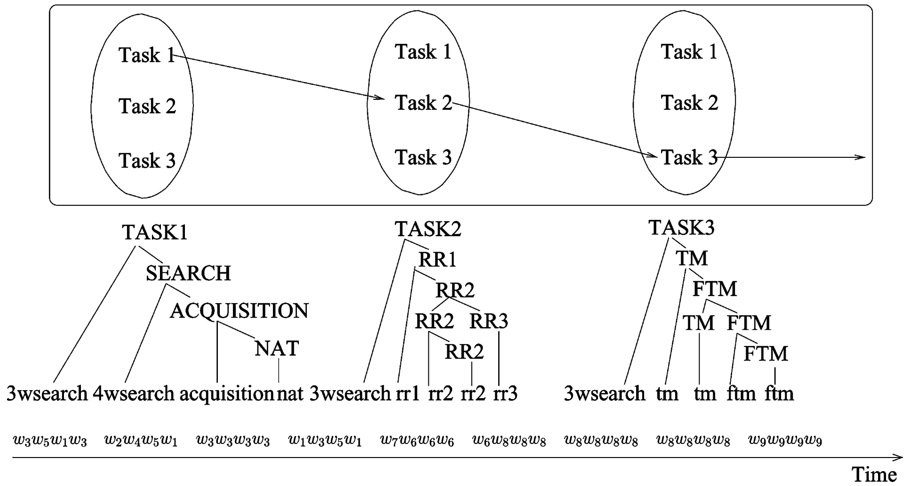


Fig. 6. A string of radar words are intercepted by the MFR, and the signal interpretation problem is, based on the domain specific knowledge on the MFR's control hierarchy, how to infer the tasks MFR is performing from the radar words. Task 1 is searching for new targets, task 2 is target identification for existing tracks, and task 3 is track maintenance for existing tracks.

C. Well Posedness of the Model

One practical issue of modeling with SCFG is that the signal generated by radar systems has finite length, and this finiteness constraint must be satisfied if the model is to be stable. In addition, the finiteness criteria provides a constraint on the SCFG model parameters, which may be used as a bound on the parameter values. We discuss this point by first defining the stochastic mean matrix.

Definition: Let $A, B \in N$, the stochastic mean matrix M_N is a $|N| \times |N|$ square matrix with its (A, B)th entry being the expected number of variables B resulting from rewriting A

$$M_N(A,B) = \sum_{\eta \in (N \cup T)^* s.t. (A \to \eta) \in P} P(A \to \eta) n(B; \eta)$$

where $P(A \to \eta)$ is the probability of applying the production rule $A \to \eta$, and $n(B; \eta)$ is the number of instances of B in η [43].

The finiteness constraint is satisfied if the grammar in each state satisfies the following theorem.

Theorem: If the spectral radius of M_N is less than one, the generation process of the stochastic context free grammar will terminate, and the derived sentence is finite.

Proof: The proof can be found in [43].

IV. Statistical Signal Interpretation of the MFR Signal and Control

Given the MFR knowledge representation as discussed previously, we are now in the position to describe the symbolic inference engine. (Recall the ES framework in Fig. 1.) The input to the engine is a batch of noisy radar words stored in a track file, and the aim is to extract the embedded syntactic pattern that is described by the domain specific knowledge. Fig. 6 illustrates the inference problem we are to solve. In general, with such an assumption, any pattern recognition technique is automatically a signal interpretation technique. Specific to our case, because the knowledge is stored as a Markov modulated SCFG, a hybrid of the inside-outside and the forward-backward algorithm will be

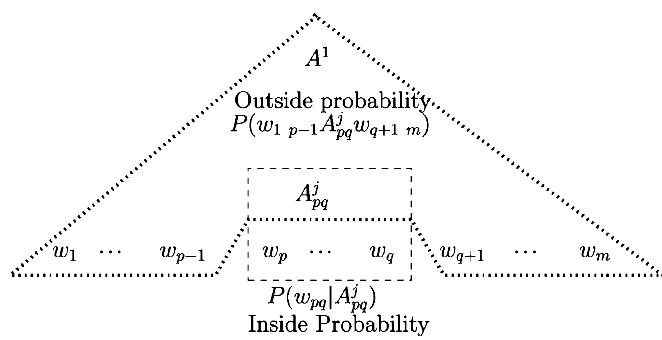


Fig. 7. Inside and outside probabilities in SCFG.

used. In this section, we describe the state estimation algorithm with the assumption of complete system knowledge (known parameter values) in Section IV-A, and the application of EM algorithm to estimate the system parameters in Section IV-B. In Section IV-C, we extend the estimation algorithm to the control of the target's maneuvering models.

Notation: The following notation will be used throughout the section. Let $x_{0:n} = (x_0, x_1, \ldots, x_n)$ be the (unknown) state sequence, where $x_k \in N_r$ (See Section III-B-1), and $\gamma_{1:n} = (\gamma_1, \gamma_2, \ldots, \gamma_n)$ be the intercepted radar commands. Each $\gamma_k = (w_1, w_2, \ldots, w_{m_k})$ is a string of concatenated terminal symbols (radar words), and m_k is the length of γ_k. It is convenient to introduce the following variables:

- forward variable: $f_i(k) = P(\gamma_1, \gamma_2, \ldots, \gamma_k, x_k = e_i)$;
- backward variable: $b_i(k) = P(\gamma_{k+1}, \gamma_{k+2}, \ldots, \gamma_n | x_k = e_i)$;
- Inside variable: $\beta^i_j(k, p, q) = P(w_{pq} | A^j_{pq}, x_k = e_i)$;
- outside variable:
  $\alpha^i_j(k, p, q) = P(w_{1(p-1)}, A^j_{pq}, w_{(q+1)m} | x_k = e_i)$

where w_{pq} is the subsequence of terminals from p^{th} position of γ_i to q^{th} position, and A^j_{pq} is the nonterminal $A^j \in N_p$ that derives w_{pq}, or $A^j \stackrel{*}{\Rightarrow} w_{pq}$. Fig. 7 illustrates the inside and outside probabilities. (Details of forward and backward algorithms can be found in [44], and inside and outside in [28].)

A. Maximum Likelihood Estimation of MFR's State via Viterbi and Inside Algorithms

The estimator of MFR's state at time k is $\hat{x}_k = \arg\max_i P(x_k = e_i|\gamma_{1:n})$, and which could be computed using the Viterbi algorithm. Define $\delta_i(k) = \max_{x_0,x_1,\ldots,x_{k-1}} P(x_0, x_1, \ldots, x_k = i, \gamma_1, \gamma_2, \ldots, \gamma_k)$, the Viterbi algorithm computes the best state sequence inductively as follows:

1) Initialization: $\delta_i(1) = \pi_i o_i(\gamma_1)$, for $1 \leq i \leq M$.
2) Induction:

$$\begin{aligned}\delta_i(k+1) &= \max_{1\leq j\leq M} [\delta_j(k) a_{ji}(\tau)]\, o_i(\gamma_{k+1}), \\ &\qquad \text{for } 1 \leq k \leq n-1, 1 \leq i \leq M \\ \tau_i(k+1) &= \arg\max_{1\leq j\leq M} \delta_j(k) a_{ji}, \\ &\qquad \text{for } 1 \leq k \leq n-1, 1 \leq i \leq M.\end{aligned}$$

3) Termination: $\hat{x}_n = \arg\max_{1\leq j\leq M} \delta_j(n)$.
4) Path backtracking: $\hat{x}_k = \tau_{k+1}(\hat{x}_{k+1})$, for $k = n-1, n-2, \ldots, 1$

where $o_i(\gamma_k)$ is the output probability of the string γ_k generated by the grammar G_i. An efficient way to calculate the probability is by the inside algorithm, a dynamic programming algorithm that inductively calculates the probability.

The inside algorithm computes the probability, $o_i(\gamma_k)$, inductively as follows:

1) Initialization: $\beta^i_j(k,p,p) = P(A^j \to w_p|G_i)$.
2) Induction:

$$\beta^i_j(k,p,q) = \sum_{r,s} \sum_{d=p}^{q-1} P(A^j \to A^r A^s) \beta^i_r(k,p,d) \beta^i_s(k,d+1,q)$$

for $\forall j, 1 \leq p < q \leq m_k$.

3) Termination: $o_i(\gamma_k) = \beta^i_1(k,1,m_k)$.

Running both the Viterbi and the inside algorithms, the posteriori distribution of the states given the observation could be computed.

B. Model Parameter Estimation Using EM Algorithm

In Section IV-A, MFR's state estimation problem was discussed assuming complete knowledge of the system parameters, i.e., the Markov chain's transition matrix and the SCFG's production rules. In reality, such parameters are often unknown. In this subsection, EM algorithm is applied for parameter estimation and it is discussed in detail in [45].

Let $\gamma_{1:n}$ be the *incomplete* data, and let $\{x_{0:n}, C_{1:n}\}$ be the *missing* (or *hidden*) data. For a Markov chain with M states, $C_k = (C^1(A \to \eta; \gamma_k), C^2(A \to \eta; \gamma_k), \ldots, C^M(A \to \eta; \gamma_k))$ and $C^i(A \to \eta; \gamma_k)$ is the number of counts the production rule $A \to \eta$ is applied in deriving γ_k with grammar i. Let $\Phi = \{a_{ji}, P^1(A \to \eta), \ldots, P^M(A \to \eta)\}$ be the model parameters, where $P^i(A \to \eta)$ is the set of production rules probabilities for grammar i. The EM algorithm iteratively computes the maximum likelihood parameter estimates by computing

$$\Phi^{(i+1)} = \arg\max_{\Phi} E_{\Phi^{(i)}} \{\log \mathcal{L}_n(\Phi)|\gamma_{1:n}, \Phi\}$$

where the complete-data likelihood $\mathcal{L}_n(\Phi)$ is $\prod_{k=1}^n P(\gamma_k, C_k|x_k, \Phi) P(x_k|x_{k-1}, \Phi) P(x_o|\Phi)$.

In order to facilitate the discussion of the EM algorithm, the following two variables are introduced

$$\chi_i(k) = P(x_k = e_i|\gamma_{1:n}) = \frac{f_i(k) b_i(k)}{\sum_{i=1}^3 f_i(k) b_i(k)}$$

and

$$\begin{aligned}\xi_{ji}(k) &= P(x_k = e_j, x_{k+1} = e_i|\gamma_{1:n}) \\ &= \frac{f_j(k) a_{ji} o_i(\gamma_{k+1}) b_i(k+1)}{\sum_{j=1}^3 \sum_{i=1}^3 f_j(k) a_{ji} o_i(\gamma_{k+1}) b_i(k+1)}.\end{aligned}$$

The Expectation step of the EM algorithm yields the following equation:

$$\begin{aligned}&E_{\Phi^{(i)}} (\log \mathcal{L}_n(\Phi)) \\ &\quad = \sum_{k=1}^n \sum_{x_k} \sum_{A^{x_k}} \sum_{T^{x_k}} E_{\Phi^{(i)}} (C^{x_k}(A \to \eta; \gamma_k)) \\ &\qquad \times \log P^{x_k}(A \to \eta) \chi_{x_k}(k) \\ &\qquad + \sum_{k=1}^n \sum_{x_k} \sum_{x_{k-1}} \log \left(a_{x_k|x_{k-1}}\right) \xi_{x_{k-1}x_k}(k-1) \\ &\qquad + \sum_{k=1}^n \sum_{x_0} \log P(x_0) \chi_{x_0}(k)\end{aligned}$$

where $E_{\Phi^{(i)}}(C^{x_k}(A \to \eta; \gamma_k))$ can be computed using inside and outside variables [35]. The Maximization step of the EM algorithm could be computed by applying Lagrange Multiplier. Since the parameters we wish to optimize are independently separated into three terms in the sum, the three terms are the estimates of the prior distribution, the transition matrix, and the production rule probabilities, we can optimize the parameter term by term. The estimates of the probabilities of the production rules can be derived using the first term of the equation, and the updating equation is

$$P^{x_k}(A \to \eta) = \frac{\sum_{k=1}^n E_{\Phi^{(i)}} (C^{x_k}(A \to \eta; \gamma_k)) \chi_{x_k}(k)}{\sum_\eta \sum_{k=1}^n E_{\Phi^{(i)}} (C^{x_k}(A \to \eta; \gamma_k)) \chi_{x_k}(k)}.$$

Similarly, the updating equation of the transition matrix a_{ji} is

$$a_{ji} = \frac{\sum_{k=1}^{n-1} \xi_{ji}(k)}{\sum_{k=1}^{n-1} \chi_j(k)}.$$

Under the conditions in [46], iterative computations of the expectation and maximization steps above will produce a sequence of parameter estimates with monotonically nondecreasing likelihood.

C. Optimization of Target-MFR Interaction Dynamics

Based on the interpretation of the radar signal and the interaction dynamics between the MFR and the target, autonomous control of the aircraft's maneuvering model is devised in this subsection. Recall the Target-MFR interaction as discussed in

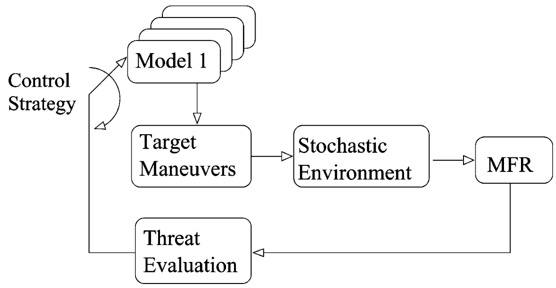


Fig. 8. The selection of maneuvering model induces a particular radar mode. The mode is observed indirectly from the intercepted radar pulses and its threat evaluated. Based on the evaluation, the control strategy selects maneuvering models such that the ownship safety is maximized.

Section III, where each maneuvering model triggers a particular radar mode, and the mode is characterized by the transition probabilities of the radar tasks. With this assumption, the maneuvering model selection is formulated as an optimization problem of finding an efficient adaptive search (sampling) plan with the objective of staying in the "safest" mode most often, and the problem setup is illustrated in Fig. 8.

Let the discrete time $l = 1, 2, \ldots$ indexes the sequence of maneuvering models selected by the aircraft. Let $X[l, s]$ be the single performance measure, the MFR's average occupancy in track mode when the target is maneuvering in model s, and which can be computed from the stationary distribution of the estimated Markov chain. The aim is to find s^* such that

$$s^* = \arg\min_{s \in \mathcal{S}} \mathbf{E}\{X[l, s]\}$$

where $\mathcal{S}$ is the set of all possible maneuvering models. The model selection is not straightforward because the performance of the maneuvering cannot be evaluated analytically, and it must be estimated or measured based on the intercepted radar pulses. We treat this problem as a discrete stochastic approximation problem. The problem is also called the multiarmed bandit where the aim is to find the best slot machine out of a finite number of such machines. Other approaches such as multiple comparison also exist [47], but this approach is preferred because of its ability to adapt to slowly time-varying radar conditions.

Two discrete stochastic approximation algorithms will be applied, and their detailed description can be found in [48]. The target begins in an arbitrarily chosen motion model, and probabilistically explore the model space. The idea is to implement an efficient adaptive sampling plan that allows one to find the maximizer with as few samples as possible by not making unnecessary observations at nonpromising models. The following notations are used in the algorithms. $\{s^{(l)}\} \in \mathcal{S}$ is a sequence of maneuvering models generated by the algorithm that can be thought as the state of the algorithm at time 1. It is convenient to map $\{s^l\}$ to a sequence of unit vectors $\{\mathbf{Y}[l]\}$ where it has 1 in the jth component if $s^{(l)} = s(j)$, and zeros elsewhere. In addition, let $\pi[l] = (1/l)[W^{(l)}[s(1)], \ldots, W^{(l)}[s(|\mathcal{S}|)]]^T$ denotes the empirical state occupation probability measure, where $|\cdot|$ gives the number of elements in the set and $W^{(l)}[s]$ is a counter that measures the number of times the state sequence visits the state s. Finally, $\hat{s}^{(l)}$ is the estimate of the optimal mode s^* generated by the algorithm at time l. It is the main output of the algorithm and it is used to control the aircraft's mode changes. The two algorithms are summarized here.

Aggressive Search:

1) Initialization: At time $l = 0$, select initial state $s^{(0)} \in \mathcal{S}$. Set $\pi[0, s^{(0)}] = 1$, $\pi[l, s] = 0$ for all $s \in \mathcal{S}$, $s \neq s^{(0)}$. Set $\hat{s}^{(0)} = s^{(0)}$.
2) Sampling and Evaluation: Given the state $s^{(l)}$, compute $X[l, s^{(l)}]$. Generate a candidate state $\tilde{s}^{(l)}$ from $\mathcal{S} - \{s^{(l)}\}$ according to a uniformly distributed random variable. Compute $X[l, \tilde{s}^{(l)}]$.
3) Acceptance: If $X[l, \tilde{s}^{(l)}] > X[l, s^{(l)}]$, then set $s^{(l+1)} = \tilde{s}^{(l)}$; otherwise set $s^{(l+1)} = s^{(l)}$.
4) Adaptive filter for updating state occupation probabilities: Update state occupation probabilities

$$\pi[l+1] = \pi[l] + u[l+1]\left(\mathbf{Y}[l+1] - \pi[l]\right)$$

with the decreasing step size $u[l] = 1/l$, where $\mathcal{Y}$ is indicator function.
5) Update estimate of optimal radar mode: If

$$\pi\left[l+1, s^{(l+1)}\right] > \pi\left[l+1, \hat{s}^{(l)}\right]$$

then set $\hat{s}^{(l+1)} = s^{(l+1)}$; otherwise, set $\hat{s}^{(l+1)} = \hat{s}^{(l)}$. Set $l \leftarrow l+1$ and go to Step 1.

Conservative Search:

1) Initialization: At frame time $l = 0$, initialize state $|\mathcal{S}|$-dimensional vectors $H[0], L[0]$ to zero, and $\bar{K}[0] = 1$ (vector of ones). Select initial state $s^{(0)} \in \mathcal{S}$.
2) Sampling and Evaluation: Given the state $s^{(l)}$, generate, as in Step 1 of Aggressive Search, $\tilde{s}^{(l)}, X[l, s^{(l)}]$, and $X[l, \tilde{s}^{(l)}]$. Update the accumulated cost, occupation times and average cost as

$$\begin{aligned} L\left[l+1, \tilde{s}^{(l)}\right] &= L\left[l, \tilde{s}^{(l)}\right] + X\left[l, \tilde{s}^{(l)}\right] \\ L\left[l+1, s^{(l)}\right] &= L\left[l, s^{(l)}\right] + X\left[l, s^{(l)}\right] \\ \bar{K}\left[l+1, \tilde{s}^{(l)}\right] &= \bar{K}\left[l, \tilde{s}^{(l)}\right] + 1, \\ \bar{K}\left[l+1, s^{(l)}\right] &= \bar{K}\left[l, s^{(l)}\right] + 1 \\ H\left[l+1, \tilde{s}^{(l)}\right] &= L\left[l+1, \tilde{s}^{(l)}\right] / \bar{K}\left[l+1, \tilde{s}^{(l)}\right] \\ H\left[l+1, s^{(l)}\right] &= L\left[l+1, s^{(l)}\right] / \bar{K}\left[l+1, s^{(l)}\right]. \end{aligned}$$

3) Acceptance: If $H[l+1, \tilde{s}^{(l)}] > H[l+1, s^{(l)}]$, set $s^{(l+1)} = \tilde{s}^{(l)}$; otherwise set $s^{(l+1)} = s^{(l)}$.
4) Update estimate of optimal radar mode: $\hat{s}^{(l)} = s^{(l)}$. Set $l \leftarrow l+1$ and go to Step 1.

The aggressive search explores the model space $\mathcal{S}$ by jumping between the models as a irreducible Markov chain, and it does not converge. However, it is shown in [48] that $\hat{s}^{(l)} \rightarrow s^*$ almost surely, meaning the algorithm spends most time at the global maximizer than any other state, and it is consistent. On the other hand, the conservative search converges almost surely to the globally optimal model. The convergence analysis of the conservative search holds for any size of the maneuvering model sequence, as long as it's greater than 0, where the aggressive search requires long sequence. In addition, one advantage of the aggressive search is that, if we keep the step size constant for both algorithms to make them adaptive to time-varying parameters, it is faster than the conservative search because it aggressively explore the state space. The numerical studies of the algorithms are discussed in the next section.

V. Numerical Studies of the Algorithms

A software testbed is implemented in C++ for MFR signal simulation and interpretation. In this section, the data structure used to implement the algorithms, and some numerical results will be discussed.

A. Implementation of the Software

The grammatical derivation process requires recursive embedding of terminals, repeated readings of nonterminals and modification of the output string. In order to have efficient repeated memory access, the production rules and their probabilities are both stored as a map data structure indexed by nonterminals, and with their right hand sides implemented with linked lists. In addition, the nonterminals and the terminals are stored as vectors, and the starting symbol as a string. With this setup, the grammatical derivation can be easily implemented by repeatedly accessing and joining the linked lists of the production rules. In addition, because any context free grammars can be reduced to Chomsky Normal Form [10], the testbed is written to accept only grammars in Chomsky normal form.

B. Model Complexity and Its Modeling Power

Here we describe briefly several implementation issues of our testbed and the possible remedies. The major implementation issue of the testbed is with the inside-outside algorithm: the computation complexity of the algorithm and the number of local maxima in the likelihood function. Suppose the MMSCFG has M states, and the states are represented by a grammar with L nonterminals. Suppose further that the observation sequence has length n, and each observation has, on average, $\hat{m}^i$ radar words for $1 \leq i \leq M$. The average case complexity of each iteration of the EM parameter estimation algorithm is $O(nM\hat{m}^3L^3)$ (The complexity of the inside-outside algorithm for radar words of length m is $O(m^3L^3)$ [35]), where $\hat{m} = E\{\hat{m}^i\}$. However, because the inside and outside algorithms could be run against the data independently, parallel computation is possible and the computation time could be reduced substantially. In order to deal with the local maxima problem, one of the approaches is to pick the initial parameter value more cleverly with pretraining method introduced in [28], where significant computational savings is recorded and EM typically converges to the global maximum.

One important implementation detail regarding the modeling power of the SCFG is its predictive power against branching processes. In [28], study is done to compare the SCFG and the HMM on their capability in modeling branching processes in terms of entropy argument. In their study, a SCFG and a HMM model are inferred against simulation data from a branching process, and it is observed that the estimated SCFG consistently has lower entropy than the estimated HMM model. Since our MFR grammar is a multitype Galton Watson branching process, SCFG has higher predictive power than HMM.

TABLE V
The Source and Estimated Parameter Values of the Markov Modulated SCFG

Source SCFG	
Grammar 1	Grammar 2
$RR1_p \xrightarrow{0.8} RR1_{p2}\ RR1_p$	$RR1_p \xrightarrow{0.2} RR1_{p2}\ RR1_p$
$RR1_p \xrightarrow{0.2} RR1_{p2}\ RR2_p$	$RR1_p \xrightarrow{0.8} RR1_{p2}\ RR2_p$
$RR1_{p2} \xrightarrow{1} RR1_c$	$RR1_{p2} \xrightarrow{1} RR1_c$
$RR2_p \xrightarrow{0.2} RR2_{p2}\ RR1_p$	$RR2_p \xrightarrow{0.8} RR2_{p2}\ RR1_p$
$RR2_p \xrightarrow{0.8} RR2_{p2}\ RR3_p$	$RR2_p \xrightarrow{0.2} RR2_{p2}\ RR3_p$
$RR2_{p2} \xrightarrow{1} RR2_c$	$RR2_{p2} \xrightarrow{1} RR2_c$
$RR3_p \xrightarrow{0.3} RR2_p\ RR3_p$	$RR3_p \xrightarrow{0.3} RR2_p\ RR3_p$
$RR3_p \xrightarrow{0.7} RR3_c$	$RR3_p \xrightarrow{0.7} RR3_c$
Transition Matrix $\begin{pmatrix} 0.7 & 0.3 \\ 0.4 & 0.6 \end{pmatrix}$	
Estimated SCFG	
Grammar 1	Grammar 2
$RR1_p \xrightarrow{0.823} RR1_{p2}\ RR1_p$	$RR1_p \xrightarrow{0.185} RR1_{p2}\ RR1_p$
$RR1_p \xrightarrow{0.177} RR1_{p2}\ RR2_p$	$RR1_p \xrightarrow{0.815} RR1_{p2}\ RR2_p$
$RR1_{p2} \xrightarrow{1} RR1_c$	$RR1_{p2} \xrightarrow{1} RR1_c$
$RR2_p \xrightarrow{0.172} RR2_{p2}\ RR1_p$	$RR2_p \xrightarrow{0.806} RR2_{p2}\ RR1_p$
$RR2_p \xrightarrow{0.828} RR2_{p2}\ RR3_p$	$RR2_p \xrightarrow{0.194} RR2_{p2}\ RR3_p$
$RR2_{p2} \xrightarrow{1} RR2_c$	$RR2_{p2} \xrightarrow{1} RR2_c$
$RR3_p \xrightarrow{0.257} RR2_p\ RR3_p$	$RR3_p \xrightarrow{0.236} RR2_p\ RR3_p$
$RR3_p \xrightarrow{0.743} RR3_c$	$RR3_p \xrightarrow{0.764} RR3_c$
Transition Matrix $\begin{pmatrix} 0.711 & 0.289 \\ 0.397 & 0.603 \end{pmatrix}$	

C. Numerical Results of the State and Parameter Estimation

In this subsection, the state and the parameter estimation algorithms derived in Section IV-A and -B are evaluated against simulation data. The model parameters such as the transition probabilities and the production rule probabilities are estimated and, based on the estimated values, the hidden state sequence is inferred. For simplicity, the MFR is characterized by a subset of the MFR grammar developed. The set of nonterminals is $\{\mathrm{RR1}_p, \mathrm{RR2}_p, \mathrm{RR3}_p\}$, and the set of terminals is $\{\mathrm{RR1}_c, \mathrm{RR2}_c, \mathrm{RR3}_c\}$. The grammars used in the numerical studies are shown in Table V in its Chomsky Normal form, and they characterize two different range resolution algorithms with different performance standards. Because the grammar is reduced, only two Markov states are considered, and the templates used to define the states are identical except their production rule probabilities. The Markov transition matrix is assumed fixed in this study. Fig. 9 shows the evolution of the likelihood values from the parameter estimation algorithm, and the state estimation error probability with the parameter values

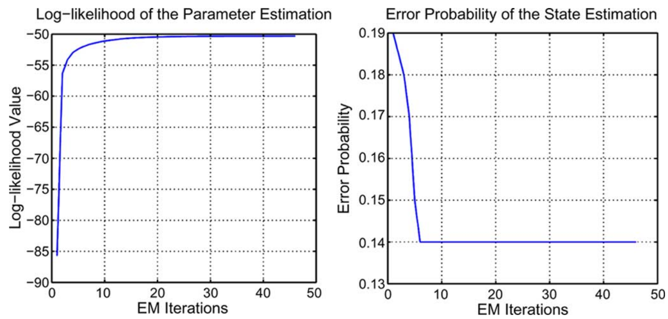


Fig. 9. The left figure shows the likelihood values obtained from iterating the parameter estimation algorithm, and the right figure is the state estimation error probability with the parameter values for each iteration of the algorithm.

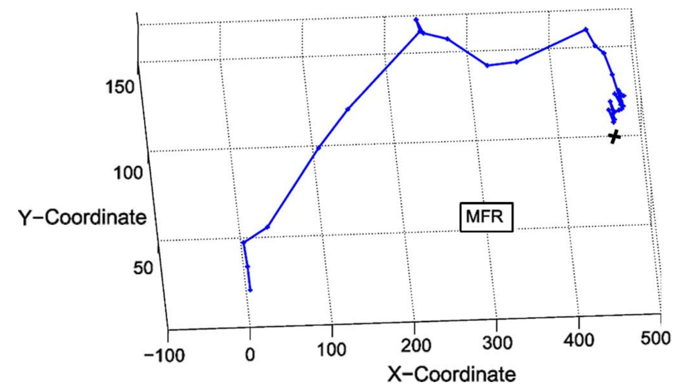


Fig. 10. The scenario of the numerical study sets a target to follow a circular path, circumventing the MFR, to reach the location labeled by X. The target's trajectory following the sequence of maneuvering models as shown in Fig. 11 is illustrated in this figure.

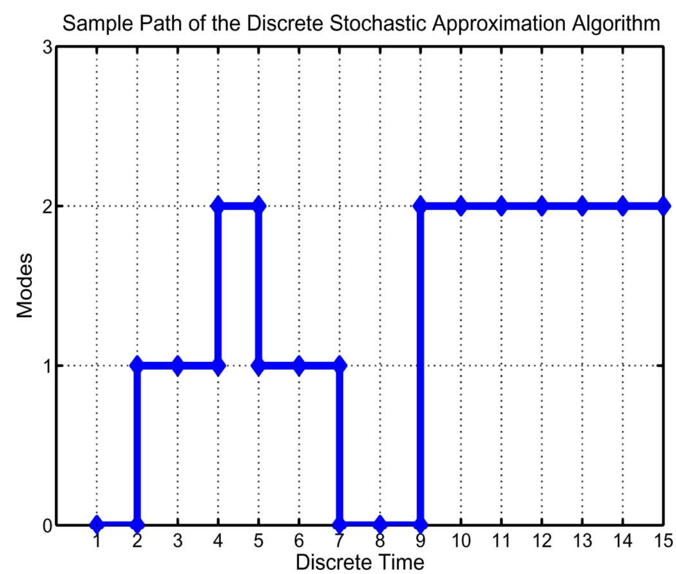


Fig. 11. The sample path of maneuvering models obtained from the discrete stochastic approximation algorithm.

for each iteration of the algorithm. The final estimated parameter values are listed in Table V, and it can be seen that the estimated parameter values are very close to their true values.

In addition, the effect of the initial values on the parameter and state estimation is also studied. We initialize the estimation algorithms with values of different square-distance from the true values, and run the parameter and state estimation algorithms. It is found that the algorithm is not sensitive to the initial values of the transition matrix, but it is sensitive to the initial values of the production rule probabilities. One observation is that if the grammars of different states are initialized too close to each other, the Markov chain degenerates into an i.i.d. sequence and the estimation algorithm updates only one state instead of two. For transition matrix along, the rms (root mean squared) error of the initial values to the true values, and of the estimated parameter values to the true model parameters are listed here. The rms error of the estimated model parameters are very close to each other despite of the differences in the initial values. Moreover, the state estimation error probabilities of the cases shown in the table at the bottom of the page all approach zero.

D. *Numerical Results of the Autonomous Selection of Maneuvering Models*

In the second numerical study, we look at the interaction between the radar and the target maneuvers, and how the target selects its maneuvering models according to discrete stochastic approximation algorithms introduced in Section IV-C. The scenario is illustrated in Fig. 10. We assume that the target intends to follow a circular path, circumventing the MFR, to reach a location labeled by X in the figure. The path is planned before the mission, and the target switches between its maneuvering models to maximize its safety.

In this paper, the target is assumed to be able to maneuver in four different motion models, and the MFR would respond with four corresponding radar modes characterized by their Markov modulated SCFG representations. Because the target's distance from the MFR stays fixed along the circular path, the MFR's transition between modes depends only on the target's maneuvering models. The SCFGs, because they correspond to the micro control, are identical across the modes (the grammar used here is the same as the one used previously), but the transition matrix of the radar manager varies depending on the target's maneuvering model. In this scenario, the simulation results from both algorithms look virtually identical, and only one set of results will be presented. Fig. 11 illustrates a sample path of the maneuvering models obtained from the algorithm, and Fig. 10 is the flight trajectory of the target following the maneuvering models. It can be seen that high maneuvering models

rms error of initial values	0.02	0.03	0.4	0.6	0.8
rms error of estimated parameters	0.14225	0.148172	0.137346	0.107406	0.144219

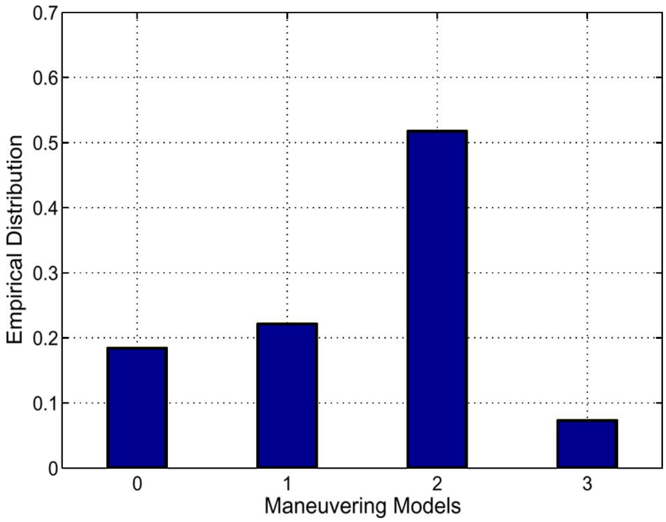


Fig. 12. Empirical distribution of the occupancies in the four maneuvering models.

are deployed at the end to ensure its survivability. Fig. 12 shows the empirical distribution of the mode occupancies after running the algorithms for 10 times, and it is observed that the maneuvering model with the highest empirical distribution is the one with the least threat, i.e., least average tracking time.

One implementation detail of the algorithm is the initialization of the Markov chain and the SCFGs. The initial parameter values are fixed for each computation of the cost function because the stochastic approximation algorithm requires the estimator to be consistent. The Markov chain is initialized uniformly, and the SCFG is initialized according to the pretraining method introduced in [28]. Briefly, the training data is first used to train a hidden Markov model with start and terminating states. The trained HMM is mapped to its approximated SCFG counterpart, and that is used as the initial configuration for the SCFG.

VI. Conclusion

The main idea of this paper is to model and characterize MFR as a string generating device, where the control rules are specified in terms of SCFG modulated by the radar's current tactical goal, and which is modeled by a Markov chain. This is unlike modeling of targets, where hidden Markov and state space models are adequate [22], [23]. The modeling is knowledge based, where each production rule corresponds to a operational rule employed by the MFR to generate its radar words, and such domain specific knowledge is assumed to be supplied by expert radar analysts. The signal interpretation of the MFR, under our formulation, is reduced to a state estimation by parsing through radar words, and a maximum likelihood sequence estimator is derived to evaluate the threat poses by the MFR. A maximum likelihood parameter estimator is also derived to infer the unknown model parameters with the Expectation Maximization algorithm. In addition, based on the interpreted radar signal, the interaction dynamics of the MFR and the target is studied and the control of the aircraft's maneuvering models is formulated as a discrete stochastic approximation problem. Since SCFGs are multitype Galton-Watson branching processes, the algorithms proposed in this paper can be viewed as filtering and estimation of a partially observed multitype Galton-Watson branching processes.

Appendix

A. A Justification of Logit Model

The Logit model can be justified by utility maximization argument. Consider only binary Logit model for simplicity, the utilities of the decisions (advancing up or down the state space as illustrated in Fig. 5) are

$$U_{\rm up} = a_{\rm u} + b_{\rm u} z_k + c_{\rm u} s_k^1 + d_{\rm u} s_k^2 + \epsilon_{\rm u}$$
$$U_{\rm down} = a_{\rm d} + b_{\rm d} z_k + c_{\rm d} s_k^1 + d_{\rm d} s_k^2 + \epsilon_{\rm d}$$

where ϵ is random threshold value. The threshold value indicates the amount of threat the MFR could take before switching of states is desired. The threshold value is random because different targets may have different threshold values. Assuming that the MFR always selects the decision with the highest utility, the probability of going up in state can be expressed as

$$\begin{aligned} P_{\rm up} &= P(U_{\rm u} > U_{\rm d}) \\ &= P\left((a_{\rm u} - a_{\rm d}) + (b_{\rm u} - b_{\rm d}) z_k + (c_{\rm u} - c_{\rm d}) s_k^1 \right. \\ &\qquad \left. + (d_{\rm u} - d_{\rm d}) s_k^2 + (\epsilon_{\rm u} - \epsilon_{\rm d}) > 0\right) \\ &= P\left(\epsilon > -\left(a + b z_k + c s_k^1 + d s_k^2\right)\right). \end{aligned}$$

Suppose that the random variable ϵ has the logistic distribution, the probability of advancing up the states, under the utility maximization argument, is expressed as

$$P_{\rm up} = \frac{\exp\left(a + b z_k + c s_k^1 + d s_k^2\right)}{1 + \exp\left(a + b z_k + c s_k^1 + d s_k^2\right)}.$$

A more general discussion for more than two states can be found in [49].

Acknowledgment

The authors would like to thank Dr. F. Dilkes and Dr. P. Lavoie of the Defense Research and Development Canada for providing useful feedback on the material presented in this paper. They would like to thank Dr. Dilkes for providing the specification of the Mercury emitter that was used as an example of modeling of a realistic MFR.

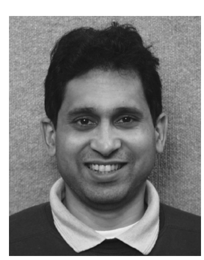
References

[1] R. G. Wiley, *Electronic Intelligence: The Analysis of Radar Signals*. Norwood, MA: Artech House, 1993.
[2] N. J. Whittall, "Signal sorting in ESM systems," in *Inst. Elect. Eng. Proc. F*, 1985, vol. 132, pp. 226–228.
[3] D. R. Wilkinson and A. W. Watson, "Use of metric techniques in ESM data processing," in *Inst. Elect. Eng. Proc. F*, 1985, vol. 132, pp. 229–232.
[4] J. Roe and A. Pudner, "The real-time implementation of emitter identification for ESM," in *Inst. Elect. Eng. Colloq. Signal Process. Electron. Warefare*, 1994, pp. 7/1–7/6.
[5] J. A. V. Rogers, "ESM processor system for high pulse density radar environments," in *Inst. Elect. Eng. Proc. F*, 1985, vol. 132, pp. 621–625.
[6] M. I. Skolnik, *Introduction to Radar Systems*. New York: McGraw-Hill, 2002.

[7] S. Sabatini and M. Tarantino, *Multifunction Array Radar System Design and Analysis*. Norwood, MA: Artech House, 1994.
[8] N. Chomsky, "Three models for the description of language," *IRE Trans. Inf. Theory*, vol. 2, no. 3, pp. 113–124, 1956.
[9] N. Chomsky and G. A. Miller, "Finite state languages," *Inf. Contr.*, vol. 1, no. 2, pp. 91–112, May 1958.
[10] N. Chomsky, "On certain formal properties of grammars," *Inf. Contr.*, vol. 2, no. 2, pp. 137–167, Jun. 1959.
[11] N. Chomsky, "A note on phrase structure grammars," *Inf. Contr.*, vol. 2, no. 4, pp. 393–395, Dec. 1959.
[12] K. S. Fu, *Syntactic Pattern Recognition and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[13] Y. A. Ivanov and A. F. Bobick, "Recognition of visual activities and interactions by stochastic parsing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 22, pp. 852–872, 2000.
[14] W. Zhu and J. Garcia-Frias, "Modeling of bursty channels using stochastic context-free grammars," in *Proc. 55th Veh. Technol. Conf. (VTC 2002)*, Birmingham, AL, May 2002, vol. 1, pp. 355–359.
[15] W. Zhu and J. Garcia-Frias, "Stochastic context-free grammars and {Hidden Markov Models for modeling of bursty channels," *IEEE Trans. Veh. Technol.*, vol. 53, no. 3, pp. 666–676, May 2004.
[16] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*, 2nd ed. Cambridge, MA: MIT Press, 2001.
[17] Y. Sakakibara, "Grammatical inference in bioinformatics," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, pp. 1051–1062, 2005.
[18] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, U,K.: Cambridge Univ. Press, 1998.
[19] E. Rivas and S. R. Eddy, "The language of RNA: A formal grammar that includes pseudoknots," *Bioinformatics*, vol. 16, pp. 334–340, 2000.
[20] M. Vilain, "Getting serious about parsing plans: A grammatical analysis of plan recognition," in *Proc. 8th Nat. Conf. Artif. Intell.*, 1990, pp. 190–197.
[21] D. V. Pynadath and M. P. Wellman, "Probabilistic state-dependent grammars for plan recognition," in *Proc. 16th Ann. Conf. Uncertainty in Artif. Intell.*, 2000, pp. 507–514.
[22] S. S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA: Artech House, 1999.
[23] Y. Bar-Shalom and X. R. Li, *Estimation and Tracking: Principles, Techniques, and Software*. Norwood, MA: Artech House, 1993.
[24] N. A. Visnevski, F. A. Dilkes, S. Haykin, and V. Krishnamurthy, "Non-self-embedding context-free grammars for multi-function radar modeling—Electronic warfare application," in *Int. Radar Conf.*, 2005, pp. 669–674.
[25] N. A. Visnevski, V. Krishnamurthy, A. Wang, and S. Haykin, "Syntactic modeling and signal processing of multifunction radars: A stochastic context free grammar approach," *Proc. IEEE*, 2007, submitted for publication.
[26] A. V. Aho and J. D. Ullman, *The Theory of Parsing, Translation and Compiling*. Englewood Cliffs, NJ: Prentice-Hall, 1972, vol. I, Parsing.
[27] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*. Reading, MA: Addison-Wesley, 2001.
[28] K. Lari and S. J. Young, "The estimation of stochastic context free grammars using the Inside-Outside algorithm," *Comp. Speech Language*, vol. 4, pp. 35–56, 1990.
[29] M. I. Miller and A. O'Sullivan, "Entropies and combinatorics of random branching processes and context-free languages," *IEEE Trans. Inf. Theory*, vol. 38, pp. 1292–1310, 1992.
[30] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. Boston, MA: Kluwer Academic, 1999.
[31] J. K. Baker, "Trainable grammars for speech recognition," in *Speech Commun. Papers for the 97th Meet. Acoust. Soc. Amer.*, 1979, pp. 547–550.
[32] D. C. Schleher, *Electronic Warfare in the Information Age*. Norwood, MA: Artech House, 1999.
[33] P. L. Bogler, *Radar Principles With Applications to Tracking Systems*. New York: Wiley, 1990.
[34] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge, U.K.: Cambridge Univ. Press, 1998.
[35] C. D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: The MIT Press, 1999.
[36] A. Wang, V. Krishnamurthy, F. A. Dikes, and N. A. Visnevski, "Threat estimation by electronic surveillance of multifunction radars: A stochastic context free grammar approach," in *Conf. Decision Contr.*, 2006.
[37] X. R. Li and V. P. Jilkov, "Survey of maneuvering target tracking. Part v: Multiple-model methods," *IEEE Trans. Aerosp. Electron. Syst.*, pp. 1255–1321, 2005.
[38] S. S. Blackman, M. T. Busch, and R. F. Popoli, "IMM/MHT tracking and data association for benchmark tracking problem," in *Proc. Amer. Contr. Conf.*, 1995, pp. 2606–2610.
[39] M. T. Busch and S. S. Blackman, "Evaluation of IMM filtering for an air defense system application," in *Proc. SPIE*, 1995, vol. 2561, pp. 435–447.
[40] R. A. Singer, "Estimating optimal tracking filter performance for manned maneuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-5, pp. 473–483, 1970.
[41] J. L. Gertz, "Multisensor surveillance for improved aircraft tracking," *The Lincoln Lab. J.*, vol. 2, pp. 381–396, 1989.
[42] N. A. Visnevski, S. Haykin, V. Krishnamurthy, F. A. Dilkes, and P. Lavoie, "Hidden markov models for radar pulse train analysis in electronic warfare," in *IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2005, pp. 597–600.
[43] Z. Chi, "Statistical properties of probabilistic context-free grammars," *Computat. Linguist.*, vol. 25, pp. 131–160, 1999.
[44] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, pp. 257–286, 1989.
[45] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Statist. Soc.*, vol. 39, pp. 1–38, 1977.
[46] C. F. J. Wu, "On the convergence properties of the algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.
[47] D. Goldsman and B. L. Nelson, "Statistical screening, selection, and multiple comparison procedures in computer simulation," in *Proc. 1998 Winter Simulation Conf.*, 1998.
[48] V. Krishnamurthy, X. Wang, and G. Yin, "Spreading code optimization and adaptation in CDMA via discrete stochastic approximation," *IEEE Trans. Inf. Theory*, pp. 1927–1949, 2004.
[49] J. S. Cramer, *Logit Models From Economics and Other Fields*. Cambridge, U.K.: Cambridge Univ. Press, 2003.

Alex Wang was born in 1979 in Taiwan. He received the B.S. degree (with honors) in engineering physics with a commerce minor and the M.S. degree in electrical and computer engineering from the University of British Columbia in 2003 and 2005, respectively.

He is currently pursuing the Ph.D. degree in statistical signal processing, under the supervision of Dr. V. Krishnamurthy, at the University of British Columbia, Vancouver, Canada. His research interests include radar signal processing, syntactic pattern recognition, and uncertain reasoning.



Vikram Krishnamurthy (S'90-M'91-SM'99-F'05) was born in 1966. He received the B.S. degree from the University of Auckland, New Zealand, in 1988 and the Ph.D. degree from the Australian National University, Canberra, in 1992.

Since 2002, he has been a professor and Canada Research Chair of the Department of Electrical Engineering, University of British Columbia, Vancouver, Canada. Prior to 2002, he was a chaired professor with the Department of Electrical and Electronic Engineering, University of Melbourne, Australia, where he also served as Deputy Head of department. His current research interests include stochastic modeling of biological ion channels, stochastic optimization and scheduling, and statistical signal processing.

Dr. Krishnamurthy has served as Associate Editor for several journals including the IEEE TRANSACTIONS ON SIGNAL PROCESSING, IEEE TRANSACTIONS AEROSPACE AND ELECTRONIC SYSTEMS, IEEE TRANSACTIONS CIRCUITS AND SYSTEMS B, and IEEE TRANSACTIONS NANOBIOSCIENCE, AND SYSTEMS AND CONTROL LETTERS. He is coeditor with S. H. Chung and O. Andersen of *Biological Membrane Ion Channels—Dynamics Structure and Applications* (New York: Springer-Verlag, 2006).