10-2013

Performance Guarantees for C-WMD Robot Missions

Shu Jiang
Georgia Institute of Technology

Ronald C. Arkin
Georgia Institute of Technology

Damian M. Lyons
Fordham University

Tsung-Ming Liu
Fordham University

Dagon Harrington
Fordham University

Performance Guarantees for C-WMD Robot Missions

Shu Jiang, Ronald C. Arkin
School of Interactive Computing
Georgia Institute of Technology
Atlanta, GA, USA

Damian M. Lyons, Tsung-Ming Liu, Dagan Harrington
Computer & Information Science
Fordham University
Bronx, NY, USA

***Abstract*—Robotics has been considered as one of the five key technology areas for defense against attacks with weapons of mass destruction (WMD). However, due to the mass impact nature of WMD, failures of counter-WMD (C-WMD) missions can have catastrophic consequences. To ensure robots' success in carrying out C-WMD missions, we have developed a novel verification framework in providing performance guarantees for behavior-based and probabilistic robot algorithms in complex real-world environments. This paper describes the system architecture and discusses how the verification framework can be used to provide pre-mission performance guarantees for robots in executing C-WMD missions.**

Keywords—performance guarantee; verification; counter weapons of mass destruction; critical missions; mobile robot; formal methods

I. INTRODUCTION

After the attack of the World Trade Center on September 11, 2001, Al-Qaeda spokesman Abu Gheith wrote that they have "the right to kill 4 million Americans" [1]. The recent bombing at the 2013 Boston marathon in the United States painfully reminded us that the safety and security of our society is under constant threats from terrorist attacks. This event reiterated the belief held by the United States military that terrorist attacks using weapons of mass destruction (WMDs) is not a question of "if" but "when" [2]. However, the resilience of human nature has fueled a history of innovation that turned tragic events into technological and scientific developments to prevent future tragic events and mitigate their effects [3].

The robotics community has been active in efforts in developing technology for use in countering terrorist threats and responding to natural disasters. The Kobe earthquake and Oklahoma City bombing motivated the development of robots for humanitarian efforts in search and rescue of trapped victims and propelled the emergence of urban search and rescue (USAR) as an important area of research for robotics [3, 4]. These efforts led to the first use of robots for search and rescue at the World Trade Center disaster in 2001 [5]. And most recently, robots were used for the Fukushima nuclear plant disaster in Japan, where the high radiation posed a substantial risk for humans to enter [6].

WMD include chemical, biological, radiological, nuclear, and high-yield explosives (CBRNE) [7]. These weapons are aimed to inflict mass casualties on a society. Thus the development of countermeasures to these weapons is imperative for safeguarding the security and safety of societies under the threat of terrorism. Typical C-WMD missions include searching, identifying, and neutralizing lethal chemical/biological agents. The time critical nature of these missions may not permit a second attempt of the mission. More importantly, with the potential mass impact of WMDs, failures to counter them effectively could have dire consequences. A C-WMD mission's success has to be ensured before execution.

As part of the United States Defense Threat Reduction Agency's (DTRA) effort to safeguard "*America and its allies from weapons of mass destruction (WMD) and provide capabilities to reduce, eliminate and counter the threat and effects from chemical, biological, radiological, nuclear, and high yield explosives*", our research addresses the challenging problem of providing performance guarantee or assurance for robots in accomplishing C-WMD missions in a real world environment. We have developed a verification-based framework to provide such guarantees for C-WMD robot missions, where a failure can have catastrophic consequences [8, 9]. The core of the framework is a verification module that conducts performance analysis of robot missions.

The verification method is based on Process Algebra [10], a mathematical tool for reasoning on process representations of robot missions. The advantage of Process Algebra is its ability to express complex robot missions and to deal with the state combinatorial explosion incurred by the robot's interaction with the environment [8]. The output of the verifier includes information beyond a simple 'yes/no' for the operator, in order to permit improvement of mission performance by modifying the control program or use of a different robot or sensors. This effectively forms a feedback loop that supports iterative improvement in the predicted mission performance.

This paper describes the verification framework and presents a C-WMD mission to illustrate how the framework can be used to verify robot mission success and its ability in dealing with complex robot missions in realistic environments. Robotic experiments of the mission are also conducted to validate the performance guarantee provided by the verification framework.

II. RELATED WORK

WMD is an extensive category, including chemical, biological, radiological, nuclear, and high-yield explosives (CBRNE) [7]. Of the WMD, biological weapons have been considered the most likely weapons of choice for terrorists [11]. Many characteristics of biological weapons make them

This research is supported by the United States Defense Threat Reduction Agency (DTRA), Basic Research Award #HDTRA1-11-1-0038.

attractive for use in terrorist attacks. Bio-weapons are colorless and odorless, which make them hard to detect. Bio-weapons are also easy to access. In 1996, an Ohio man with connections to an extremist group was able to obtain bubonic plague cultures through the postal service [12]. Recipes for making biological weapons are even available online [12]. Bio-weapons are also characterized by easy delivery, low production cost, lethal in low dose, easy transportation, and potentially contagious (e.g., smallpox) [13].

Robotics has been considered as one of the five key technology areas for defense against attacks with weapons of mass destruction (WMD) [14]. Humphrey [15] had identified eight C-WMD tasks for robots, which include survey, identification, scene observation/object tracking, medical initial assessment, medical victim transportation, decontamination, hazard disposal, and resource hauling. However, to successfully deploy robots to accomplish these tasks autonomously or semi-autonomously in a real-world environment still remains a great challenge for robotics. As tasks increase in complexity, so do the robotic systems that are designed to do these tasks. When coupled with the environment, increased complexity in both the task and robotic systems increase the number of ways the systems can fail [16].

The time-critical and mass impact nature of C-WMD missions does not tolerate failures. Poor judgments of the robot's abilities to operate in the real world have caused the failures of many robotic systems [16]. Thus, it is the goal of our research to develop tools that can be used to generate performance guarantees of robotic systems for these tasks and to aid the mission commander/robot operator in decision making regarding the usage of robots. This paper presents our software framework, based on formal methods, in providing performance guarantees for behavior-based and probabilistic robot algorithms in complex real-world environments.

Formal methods for robots have recently emerged as an important area of research in robotics. This is driven by the increasing need to guarantee the safety and correct behavior of robotic systems (e.g., surgical robots). Formal methods are mathematical tools for verification and synthesis of software and hardware systems [17]. Major progress has been made in tools for verification of software and hardware systems. However, robotics presents new challenges for formal methods due to the fact that robots have to continuously interact with the environment, which can be unstructured, uncertain, and dynamic.

Formal verification methods for robotics can be generally classified into two major categories: synthesis and verification. Synthesis deals with the problem of automatically generating correct-by-construction controller for a robot or team of robots, given a model for the robot and its specification expressed in a formal language, such that the robot is guaranteed to satisfy the given specification [17, 18]. Most synthesis approaches use Linear Temporal Logic (LTL) as the formal specification language [16, 19-23]. However, it is not clear that LTL is the right specification language [24]. Other specification languages such as Computational Tree Logic (CTL) [25] and Interval Temporal Logic (ITL) [26] can be viable alternatives to LTL.

Verification addresses the problem of proving the correctness of a control system with respect to a formal specification or property using formal methods [16, 27]. Model checking has been a widely used technique for this purpose [16, 27-30]. In model checking, the system is represented as a finite state automaton (FSA) and formal specifications are verified by exhaustive exploration of the system's reachable states [16, 28]. However, these methods suffer from the well-known combinatorial explosion problem. Model checking has been extended to deal with stochastic systems using statistical sampling methods [31-33]. Statistical methods reduce the cost of probabilistic model checking by replacing numerical computation of probabilities with sampling and require minimum memory by avoiding model construction. However, with sampling, there is no guarantee that the verification procedure always produces the correct answer [32].

Our ongoing research in formal verification of robotic systems focuses on verifying performance guarantees for a robot or a team of robots in carrying out C-WMD missions in real environments. The fundamental problem for verifying robot behavior is the interaction between the robot and the environment, which might cause unpredictable behaviors to emerge. Thus, to verify performance guarantees about the robot behavior, we have to model the interactions between the environment and the robot to the extent to which it is known. We have developed a verification framework, VIPARS (*Verification In Process Algebra Robot Schemas*), to represent robots, sensors, the control programs, and the operating environments [8, 9]. Process algebra enables verification of the system behavior (i.e., composition of processes) through automated algebraic reasoning as described in the following section.

III. System Architecture

Inserting robots into the C-WMD team not only changes the nature of the countermeasure mission, but also the team dynamics and its decision making (e.g., task allocation). One critical decision that the mission commander needs to make is whether robots should be used for a particular C-WMD mission, where the failure of the mission can be catastrophic. Such a decision hinges upon the predicted performance of the robot in carrying out the mission. It is the objective of our research to assist the mission commander in making this decision by providing information regarding performance guarantees using robots in accomplishing C-WMD missions in the real world. This section describes the framework for providing such a guarantee.

A. Overview of the Framework

The verification-based system for providing performance guarantee is built upon *MissionLab*, a behavior-based robot programming environment [34] (Fig. 1). The front-end of the *MissionLab* programming environment is a usability-tested graphical robot behavior programming interface, where the robot program is created as a finite state automaton (FSA). *MissionLab* provides a library of primitive behaviors (e.g.,

obstacle avoidance, go to goal, etc) that can be assembled into higher-level complex behaviors (e.g., a biohazard search behavior) in the form of FSAs. The newly added verification system is intended to provide the robot operator with the additional capability of verifying the performance of the robot program she constructed for the mission at hand prior to deployment.

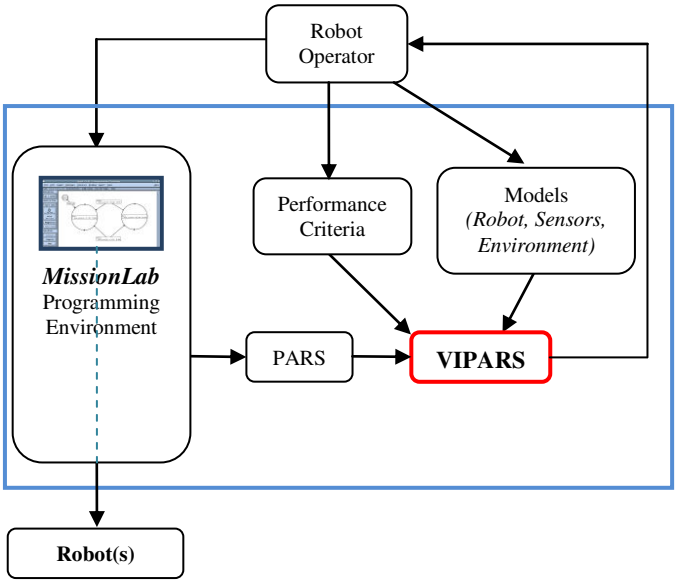

Fig. 1. System Architecture

The core of the proposed system in Fig. 1 is the process algebra-based verification module named VIPARS (*Verification in Process Algebra Robot Schemas*) [8]. To obtain the performance guarantee for a mission, the robot program is translated to PARS (*Process Algebra Robot Schemas*), the specification language of VIPARS. The robot operator also needs to provide VIPARS with models of the robot, the sensors it is equipped with, and the operating environment of the mission, typically from a pre-existing library. Lastly, the performance criteria that the mission is required to meet need to be specified as well. VIPARS then provides the operator with the performance guarantee for the mission based on how well the provided performance criteria are satisfied by the given control algorithm, robot, and the operating environment.

Based on the output of VIPARS, the robot operator can decide whether to abort the mission due to low confidence on success or to modify elements of the mission to improve performance (e.g., use a different robot or modify the control algorithm). The performance guarantee of a mission's success is quantified as a probability distribution that describes the likelihood of a robot successfully carrying out the mission by satisfying all specified performance criteria [35], which can be thresholded to provide a binary 'yes/no' answer to the operator if desired. Additionally, the verification module provides the human operator with other information that is useful for iterative improvement of the mission performance (e.g., unsatisfied constraints, probabilities of meeting each criterion).

The verification module effectively forms a feedback loop that the human operator can iteratively improve the predicted performance of a robot in executing a mission. By providing pre-mission performance analysis of robots in carrying out C-WMD missions, the verification system under development can assist a human robot operator in making critical decisions regarding the deployment of robots. With the knowledge of what would work and what would fail, the mission commander/robot operator could save valuable time and avoid catastrophic consequences by avoiding mission attempts that can lead to failure.

B. VIPARS

Previously, as a technique for general-purpose software verification, model checking has had very good results [36]. A program is converted to a state transition system in which each state is labeled by a set of propositions on program variable values. For robot program verification, however, we must not only analyze the robot program but also the variables associated with the model of the physical environment. The combination of environmental model and program (both of which have concurrency, uncertainty and continuous-valued variables) renders a state-space prohibitively large to use reachability (and thus traditional model checking) as a practical verification paradigm.

A standard approach to deal with this state-space explosion is to search for state regularities that can be leveraged to reduce the size of the space: for example, counterexample guided predicate abstraction [37] and abstract interpretation [38]. In its interaction with the physical environment, a behavior-based robot will regularly respond to a fixed set of sensory percepts. This induces a periodic regularity in the combined state-space of program and environment. While the robot may transition to another behavioral state with its own sets of percepts (behaviorally-relevant sensor abstractions), thereby adding complexity to the interaction between robot and environment, it is this repeated handling of a specific set of percepts that is leveraged to make the verification problem tractable. This notion of periodic regularity will be more precise after a brief introduction to PARS.

In PARS, a process P with initial parameter values u1, u2 … input ports/connections i1, i2 ... output ports/connections o1, o2 ... and final result values v1, v2 ... is written as:

$$\mathbf{P}_{u1,\,u2\,\ldots}\,(i1, i2\ldots)\,(o1, o2\ldots)\,\langle v1, v2\ldots\rangle \qquad (1)$$

If there are parts of a process description that are empty, they are typically omitted. Processes that are defined only in terms of a port-automaton are the atomic units, or basic processes, from which the programs are built. Examples of basic processes are shown in Table I.

TABLE I. EXAMPLES OF BASIC PROCESSES

Process	Stop	Abort
$\mathbf{Delay}_t$	After time t	If forced
$\mathbf{Ran}_{\Phi}\langle v\rangle$	returns a random sample v from a distribution Φ	If forced
$\mathbf{In}_c\langle y\rangle$, $\mathbf{Out}_{c,x}$	perform input and output, respectively, on port c	If forced
$\mathbf{Eq}_{a,b}$, $\mathbf{Neq}_{a,b}$, $\mathbf{Gtr}_{a,b}$, etc	a=b, a!=b, a>b, etc	Otherwise

Non-basic processes are defined in terms of compositions of other processes. For example, a process T that inputs a value on port c1 and then outputs it on port c2 is defined:

$$\mathbf{T} = \mathbf{In}_{c1}\langle x\rangle \; ; \; \mathbf{Out}_{c2,x} \quad (2)$$

, where ';' denotes sequential and conditional composition. Other composition operations include parallel-max (|) and parallel-min or disabling (#). The iterative construct in PARS is recursion. A tail-recursive (TR) process is written as:

$$\mathbf{T}_a = \mathbf{P}_a\langle b\rangle; \mathbf{T}_{f(a,b)} \quad (3)$$

This describes a process that repeats P until it aborts. Any language that implements sequence, condition and loop constructs is sufficient to represent any program [39]; thus, we are confident that PARS can represent any robot program.

In [9], we introduced the System Period as a state regularity to address the significant combinatorial problems of a state-based approach. The System Period can be roughly described as follows: given a System that is the concurrent, communicating composition of robot controller and environment (which is expressed in PARS as a parallel composition of TR processes: **Sys = P1 | P2 | ... | Pn**), we would like to rewrite it in TR form for the purposes of verification. First, we define the 'period' (Pi') of a TR process, P, as the section of the definition between the equal sign and the tail-recursive call. This yields an expression of the form **Pi = Pi' ; Pi**. If all of the periods in the component processes of Sys (**P1', P2', ..., Pn'**) contain port communication and all of the input and output communications can be matched (some periods will need to be unrolled), then we can specify a period for the entire System : **Sys = Sys' ; Sys**. The above requirement that all input and output communication can be matched solves the classical deadlock problem. A second constraint that we place on the communications between component processes is that no more than two processes are ready to communicate on the same port at the same time; hence starvation is a non-issue as no competing processes are being denied their voice on an input-output channel, for example. This is done to simplify the computational complexity of the generation of the System Period without compromising the representational ability - multiple inputs on a port just need to be explicitly sequenced now. Under these constraints, the isolation of this period turns out to be the formal identification of the regularity in behavior-based programs.

Taking advantage of the behavioral system period, the VIPARS verification module generates the System Period [9] by analyzing the recurrent structure of the concurrent composition of the robot mission controller and an environment model. VIPARS analyzes the port connectivity of processes within the period to determine the way in which process variables are transformed, thereby producing a set of recurrent functions we call flow-functions. VIPARS then, upon specification of initial variable values and goal variable values, attempts to solve these flow-functions with these boundary conditions using a Dynamic Bayesian Network approach [40].

VIPARS verifies a performance guarantee for a given robot mission where the models of the robot, sensors, and the environment are specified by the robot operator. For example, consider a single-waypoint mission, where the robot controller attempts to move the robot from point P_0 to a goal point G. Capturing the uncertainty associated with sensor and actuator performance, the process parameter for robot position is represented as a distribution of positions. At each time-step through the Bayesian network, VIPARS evaluates sensor and actuator performance probabilities and calculates the probability that mission criteria are met. The robot operator needs to specify T_{max} and P_{min} as performance criteria; that is, the measure for success is that the robot must be at the goal point G after some time $t<T_{max}$ with a probability $p>P_{min}$.

The output of VIPARS is two-fold: 1) a Boolean answer to whether the mission can be successful and 2) a detailed record of the values of variables and distributions through successive time steps. From this output, a feedback loop is created which gives the robot operator the ability to refine the robot program until they are sufficiently confident to deploy the robot.

IV. Verification of a C-WMD Mission

A typical scenario that motivates the development of our C-WMD missions is the sarin gas attack of the Japanese subway system [12]. On March 20, 1995, members of the Japanese cult Aum Shinrikyo released sarin gas, a lethal nerve agent, on the subway trains in Tokyo. This attack resulted in 13 deaths and thousands of injuries. Sarin is colorless and odorless, which made it undetected by victims until symptoms started to appear. Some injuries were preventable, but delays in identifying the responsible agent allowed contamination to spread to hospitals, where staff failed to put on protective clothing and gas masks [12]. This event highlights the need to develop effective countermeasures for this type of attack.

The key element of a countermeasure to a biological attack is the rapid identification of the biological agent used before the agent is widely disseminated [11]. Only after the nature of the biohazard is identified, can the first responders proceed to decontaminate the hot zone area and administer appropriate treatment for victims. In this section, we present a *Biohazard Search* mission where a robot is tasked to search an area for biohazard, Fig. 2.

Fig. 2. Indoor Biohazard Search

The control program of the robot for the mission, as shown in Fig. 3, is constructed in *MissionLab* as a behavioral

assemblage in the form of a FSA. The FSA consisted of three behaviors (*Wander, MoveToward, and Stop*) and three triggers (*Detect, NotDetected, and Near*). With this behavioral assemblage, the robot starts with random exploration of the environment. However, when *Detect* is triggered, the robot switches from random exploration to moving toward the detected biohazard. This mission is completed once the robot is within a certain distance of the biohazard. While we adopted a simple search strategy (i.e., *Wander* behavior) for this mission, more sophisticated search strategies can be employed as well. VIPARS can potentially be used by the human operator to verify which search strategy would be most effective.

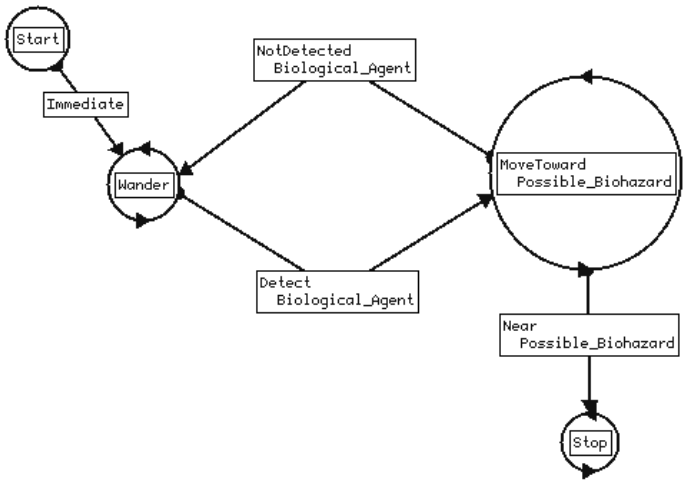

Fig. 3. FSA of the Biohazard Search Mission

When the robot operator is finished with the design of the robot behavior, the control program is then translated into PARS, the language of the VIPARS, for verification. The PARS representation of the *Biohazard Search* behavior in Fig. 3 is expressed as

Behavior = **NotDetected ; (Detected # Wander) | Detected ; (Near # MoveToward) | Near ; Stop** (4)

The above can be described as the concurrent composition of the following behaviors: 1) if not detected, wander until found, 2) if detected, move until near and 3) if near, stop. The **Behavior** process models the high-level *Biohazard Search* behavior. The **Behavior** process consists of a hierarchically nested process network. The lower-level processes in the network are PARS models of primitive behaviors, which are translated offline to PARS and stored in a library. This follows the spirit of behavior-based robotics [41] and simplifies the representation of complex behaviors in PARS.

Different missions have different requirements that the robot has to meet. For the *Biohazard Search* mission, we are interested in time performance, successful detection of biohazard, and correct identification of the biohazard. These performance criteria are expressed in PARS as a performance specification network:

Criteria = **Delay**⟨t⟩ **; (At**⟨p⟩ **| Biohazard**⟨p⟩ **)** (5)

The **At**⟨p⟩ process indicates the robot is at location p, the **Biohazard**⟨p⟩ process indicates the location of the biohazard, and the **Delay**⟨t⟩ process indicates a time t has passed. In verification we ask whether the mission will achieve this liveness condition for $t<T_{max}$ with at least probability P_{min}. The **Criteria** process is a property specification process network, which differs from a process network in that it is actually a process network constraint expression, a specification of a set of possible networks [9].

A robot model generally includes the kinematic model of the robot platform. More complex robot models would include robot dynamics, battery life, tire properties, etc. The robot to be used for this mission is a Pioneer 3-AT, Fig. 4, and it is modeled in PARS as:

Robot = (**Delay**⟨t⟩# **Odo**⟨r⟩ # **At**⟨r⟩) ; **Ran**⟨Θ⟩⟨z⟩ ; **In**⟨v⟩⟨u⟩ ; **Robot**⟨r+(u+z)t ⟩ (6)

Odo = **Ran**⟨Φ⟩⟨e⟩ ; **Out**⟨p,r+e⟩; **Odo**⟨r⟩ (7)

The process $\mathbf{At}_r$ represents the robot at location *r*. The process **Odo** (short for Odometry sensor) transmits the robot location in a loop until terminated by the **Delay**. In our results, we use a multivariate normal distribution for these distributions. The sensor noise is characterized by the distribution $\Theta \sim N(\mu s,\sigma s)$ and actuator noise by $\Phi \sim N(\mu m,\sigma m)$.

Fig. 4. Pioneer 3-AT

We chose to separate some sensor models from robot model because the same robot platform can be equipped with different external sensors. By separating certain sensor models from robot models, it allows the addition/change of sensors without modifying the robot models. For this mission, the Pioneer 3-AT robot is equipped with a camera for biohazard detection and a SICK laser scanner for obstacle avoidance. The sensor model is a composite model of these sensors, which can be expressed in PARS as:

Sensor = **Camera | Laser** (8)

Camera = **Out**<s,sv> ; **Camera** (9)

Laser = **Out**<o,svo> ; **Laser** (10)

The **Out**<s,sv> and **Out**<o,svo>processes output current target sensor information and current obstacles respectively.

The fundamental problem for the verification of robot behavior is the interaction between the robot and the

environment, which can be uncertain, unstructured, and dynamic. Undesirable robot behaviors might emerge through this interaction, which might not have been foreseen by the robot programmer/operator. Thus, it is important for a verification language to be expressive enough to represent complex real-world environments. The targeted environment for the *Biohazard Search* mission is an indoor environment, Fig. 2. The PARS model of the environment is expressed as:

Environment = (**Out**⟨pi,qi⟩ # **Out**⟨po,qo⟩ #**In**⟨p⟩⟨q⟩) ; (11)
(**Cond**⟨inside,q⟩⟨qi⟩ |
Cond⟨outside,q⟩⟨qo⟩) ;
Environment⟨qi,qo⟩

Random variable values, such as the robot position, are represented in PARS as Gaussian Mixtures [40].The **Environment** process tests the robot position probability distribution and separates into two mixtures: on representing the portion of the distribution that is inside the room (*qi*), and one that represents the portion that would collide with the room walls (*qo*). That portion will be channeled to the **Camera** and **Laser** sensor processes to return information back to the control strategy.

The PARS models of the control program, robot, sensors, and the environment form the System process, which is the concurrent, communicating composition of component processes

System = **Behavior | Environment | Robot | Sensor** (12)

The **System** process is then analyzed by VIPARS to determine if it satisfies all the constraints specified by the property specification process network (i.e., the **Criteria** process).

V. RESULTS

This section presents the result of the verification system for the *Biohazard Search* mission, which tasks a robot to search a room for a potential biohazard within a certain time limit. Experimental validation of the mission is also conducted to show the validity of the verification result. Validation basically consists of running the mission on a physical robot in the real environment and measuring the robot's performance. Validation and verification results are compared using a z-statistic proportion test to determine if any statistical significant difference exists between these results.

A. Verification of Biohazard Search Missions

VIPARS performed the verification of the *Biohazard Search* mission to provide a performance guarantee of the mission for the robot operator in a given environment. The mission is verified against two performance criteria: 1) locate the biohazard successfully and 2) complete the mission within 60 seconds. The biohazard is successfully located by the robot when it is within 1.0 meter of the biohazard. The mission is completed when the biohazard is located by the robot; however, the mission is only successful when the biohazard is located within 60 seconds since the start of the mission. The operating environment of the mission is assumed to be a medium-sized room (i.e., 10m x 10m) for verification. Another assumption made for verification is that the biohazard is assumed to be in the room and its probability of being at any point in the room is considered to be uniformly distributed. The environment is also assumed to be free of obstacles. However, the experiment can be extended to study a clustered environment where the biohazard might be hidden behind some obstacles. Introducing obstacles into the environment is expected to increase the search time and decrease the probability of mission success.

With the mission represented by the **System** process (12), VIPARS first generates the system period and flow functions for the process with the *SysGen* and *FloGen* algorithms we developed in prior work [8, 42]. A parameter flow function is a mapping $f : \mathbb{D}^m \rightarrow \mathbb{D}^m$ that relates the values of m parameters (i.e., process variables) in the n^{th} and $(n+1)^{th}$ iterations of the system period [9]. For example, the robot's position is updated at each iteration of the recursive process using the flow function:

$$f(p_{t+1}) = p_t + vt \quad (13)$$

where p is the robot position, v is the robot velocity, and t is the time step. We have shown that verification then consists of solving these flow functions for the initial variable values and the goal variable values as boundary conditions [42]. We also developed a Dynamic Bayesian Network (DBN) approach for solving these flow functions [42].

Our approach of evaluating the mission success consists of two stages, in which first one is to determine the probability that the robot will detect the target given the camera sensitivity data. The second stage is basically simply evaluating the probability of success of the robot moving toward the target, within the time left in the time constraint that has not been used up in the first stage. The camera calibration model was given as a probability of target detection given the target was y meters from the robot for y = 1m to 10m; we write this as $P(s|target_{at_y}, robot_{at_0})$. Only the mean data was used and a Gaussian was fit to the data based on minimizing the error between the Gaussian and the calibration data in the range 1m to 10m. Target detection probability at time t was expressed as

$$P(s) = \sum_x \sum_y P\left(s\middle|target_{at_{x+y}}\right) P(robot_{at_x}) P\left(target_{at_y}\right) \quad (14)$$

where $P\left(target_{at_y}\right)$ is the same at every point (i.e., the target can be located anywhere in the room) and sums to 1 for the room. The robot position distribution $\mathrm{P}(\mathrm{robot}_{\mathrm{at_x}})$ is convolved with the zero-mean camera calibration $\mathrm{P}(\mathrm{s}|\mathrm{target}_{\mathrm{at_y}}, \mathrm{robot}_{\mathrm{at_0}})$ to get $\mathrm{P}\left(\mathrm{s}\middle|\mathrm{target}_{\mathrm{at_{x+y}}}\right)\mathrm{P}(\mathrm{robot}_{\mathrm{at_x}})$ and this is numerically integrated over the room to get the probability of detection at this discrete time. The probability of detection on each time step is also assumed independent.

The performance guarantee of a mission is the quantification of the ability of the specified mission to be completed successfully in the given environment [35]. Currently, we represent this quantity as the probability of mission success with respect to the specified performance criteria. The result of the verification is summarized in Table II. The result indicates that the verifier predicted an 85%

success probability for the *Biohazard Search* mission with the robot operating in the environment with respect to the performance criteria specified earlier. This result can be thresholded to get a 'yes/no' answer for the robot operator regarding whether the mission will be successful. This information can then be used by the operator to make the decision of whether to deploy the robot or not.

TABLE II. VERIFICATION RESULT[1]

Mission	Performance
Biohazard Search	85.0 %

B. Validation

Rigorous experimental validation needs to be conducted to validate the predicted performance generated by verification. For the *Biohazard Search* mission, validation experiments were carried out to obtain the actual performances of the robot in the real environment. For the validation experiment, the operating environment of the robot is a room with a dimension of approximately 10m x 12m, Fig. 2. The room is covered with tile flooring and is well lit by florescent lights. The major area of the room is empty except some standard items along the walls (e.g., cabinets, storage crates).

The robot that was used for this mission is a Pioneer 3-AT, Fig. 4. The robot is equipped with a laser scanner for obstacle avoidance and a forward-facing camera for biohazard detection. The camera has a field of view of 39.6 degrees. The biohazard is represented by a red biohazard bucket, Fig. 4. The bucket is 0.38m in height and 0.3m in diameter. Color feature of the biohazard bucket is used for biohazard detection.

The complete validation experiment consists of 106 trial runs of the *Biohazard Search* mission. For each trial, the robot starts at the entrance of the room and proceeds to search the room with the control program described in Fig. 3. For all the trials, the location of the biohazard is uniformly distributed with respect to the room. Each trial is completed when the robot locates the biohazard. Mission success is defined by the performance criteria. For this mission, the criterion is that the robot needs to find the biohazard in 60 seconds. Thus, the time it takes for the robot to locate the biohazard is recorded for each trial. The result of the validation experiment is shown in Table III.

TABLE III. VALIDATION RESULT

Mission	# Trials	# Successes	Performance
Biohazard Search	106	88	83.0 %

[1] The initial verification result was reported blindly as 79% without the knowledge of the validation result, which resulted in $z = 1.01$ and $P(Z<1.01) = 0.16 > 0.05$. However, it was discovered that this initial result used a stopping radius of 0.75m rather than 1.0m. The verification was rerun with the correct condition, after the initial comparison between verification and validation results, which resulted in the corrected 85% value.

C. Comparison of Verification and Validation Resutls

Verification of *Biohazard Search* mission predicted an 85% mission success probability, while the validation experiments showed an actual robot succeeds 83% of the time based on 106 trials with 18 failures. Validation and verification results are compared using a z-statistic proportion test to determine if any statistical significant difference exists between these results. The null hypothesis is H_0: $p_{succ}=0.85$, and the alternative hypothesis is H_a: $P_{succ}<0.85$. The z-statistic for the results is calculated in (15).

$$z = \frac{p_1 - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}} = \frac{0.83 - 0.85}{\sqrt{\frac{0.85(1-0.85)}{106}}} = -0.58 \quad (15)$$

We obtained a z-statistic as z = -0.58, which resulted in P(Z<-0.58)= 0.28 from the standard distribution table. Since 0.28>0.05, we fail to find any statistically significant difference between the verifier's performance prediction and the actual performance from the validation experiments. We are then safe to conclude that the VIPARS' performance guarantee, the 85% probability of mission success with respect to the performance criteria, for the *Biohazard Search* mission is a valid prediction.

VI. CONCLUSION

The goal of our research is to develop a tool for automatic verification of performance guarantees of a robot or a team of robots for critical C-WMD missions in complex real-world environments. To this end, we have been developing a formal verification framework, named VIPARS, based on process algebra, which allows algebraic reasoning over the PARS models of the system. Given the performance criteria for a C-WMD mission, VIPARS verifies whether the combination of control program, robot, sensor, and environment models will satisfy the mission criteria. The output of VIPARS is also designed to provide guidance to an operator to improve mission performance if the initial predicted performance is not satisfactory. This paper also presented a C-WMD mission that was used to illustrate how the verification framework can be used to provide performance guarantees of robots operating in real-world environments. Experimental validation of the *Biohazard Search* mission was conducted to obtain the actual performances of the robots. A comparison between the verification and validation results showed the proposed system's ability in providing valid performance guarantee for time-critical missions.

Future work includes verification of additional C-WMD missions such as multi-robot coordination, outdoor building approach, and simultaneous localization and mapping (SLAM) in unknown environments. More extensive verification and validation of robot performance will also be conducted at the USAR test facility at the National Institute of Standards and Technology (NIST) [43].

REFERENCES

[1] M. D. Intriligator and A. Toukan, "Terrorism and weapons of mass destruction," *Countering Terrorism and WMD: Creating a Global Counter-terrorism network,* 2006.

[2] L. E. Dickinson, "Military Role in Countering Terrorist Use of Weapons of Mass Destruction," No. AU/AWC/093/1999-04 Air War College, Maxwell Air Force Base, Alabama, 1999.

[3] A. Davids, "Urban search and rescue robots: from tragedy to technology," *Intelligent Systems, IEEE,* vol. 17, pp. 81-83, 2002.

[4] H. Kitano and S. Tadokoro, "Robocup rescue: A grand challenge for multiagent and intelligent systems," *AI Magazine,* vol. 22, p. 39, 2001.

[5] J. Casper and R. R. Murphy, "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics,* vol. 33, pp. 367-385, 2003.

[6] K. Nagatani, S. Kiribayashi, Y. Okada, K. Otake, K. Yoshida, S. Tadokoro*, et al.*, "Emergency response to the nuclear accident at the Fukushima Daiichi Nuclear Power Plants using mobile rescue robots," *Journal of Field Robotics,* vol. 30, pp. 44-63, 2013.

[7] L. D. Prockop, "Weapons of mass destruction: overview of the CBRNEs (chemical, biological, radiological, nuclear, and explosives)," *Journal of the neurological sciences,* vol. 249, pp. 50-54, 2006.

[8] D. M. Lyons, R. C. Arkin, P. Nirmal, S. Jiang, and T.-M. Liu, "A Software Tool for the Design of Critical Robot Missions with Performance Guarantees," in *Conference on Systems Engineer-ing Research (CSER'13)*, Atlanta GA, 2013.

[9] D. M. Lyons, R. C. Arkin, P. Nirmal, and S. Jiang, "Designing autonomous robot missions with performance guarantees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 2583-2590.

[10] J. C. Baeten, "A brief history of process algebra," *Theoretical Computer Science,* vol. 335, pp. 131-146, 2005.

[11] D. A. Henderson, "The looming threat of bioterrorism," *Science,* vol. 283, pp. 1279-1282, 1999.

[12] R. Danzig and P. B. Berkowsky, "Why should we be concerned about biological warfare?," *Jama,* vol. 278, p. 431, 1997.

[13] E. J. DaSilva, "Biological warfare, bioterrorism, biodefence and the biological and toxin weapons convention," *Electronic Journal of Biotechnology,* vol. 2, pp. 3-4, 1999.

[14] M. J. C. Doesburg and C. General, "The Evolution of Chemical, Biological, Radiological, and Nuclear Defense and the Contributions of Army Research and Development," NBC Report, the United States Army Nuclear and Chemical Agency, Fall / Winter 2004.

[15] C. M. Humphrey and J. A. Adams, "Robotic tasks for chemical, biological, radiological, nuclear and explosive incident response," *Advanced Robotics,* vol. 23, pp. 1217-1232, 2009.

[16] J. M. Braman, "Safety verification and failure analysis of goal-based hybrid control systems," Doctoral Dissertation, California Institute of Technology, 2009.

[17] R. van de Molengraft, M. Beetz, and T. Fukuda, "Robot challenges: Toward development of verification and synthesis techniques," *Robotics & Automation Magazine, IEEE,* vol. 18, pp. 22-23, 2011.

[18] H. Kress-Gazit, T. Wongpiromsarn, and U. Topcu, "Correct, reactive, high-level robot control," *Robotics & Automation Magazine, IEEE,* vol. 18, pp. 65-74, 2011.

[19] H. Kress-Gazit and G. J. Pappas, "Automatic synthesis of robot controllers for tasks with locative prepositions," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 3215-3220.

[20] X. C. Ding, M. Kloetzer, Y. Chen, and C. Belta, "Automatic deployment of robotic teams," *Robotics & Automation Magazine, IEEE,* vol. 18, pp. 75-86, 2011.

[21] M. Lahijanian, S. B. Andersson, and C. Belta, "Temporal logic motion planning and control with probabilistic satisfaction guarantees," *IEEE Transactions on Robotics,* vol. 28, pp. 396-409, 2012.

[22] V. Raman, N. Piterman, and H. Kress-Gazit, "Provably Correct Continuous Control for High-Level Robot Behaviors with Actions of Arbitrary Execution Durations," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[23] E. M. Wolff, U. Topcu, and R. M. Murray, "Efficient reactive controller synthesis for a fragment of linear temporal logic," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[24] C. Belta, A. Bicchi, M. Egerstedt, E. Frazzoli, E. Klavins, and G. J. Pappas, "Symbolic planning and control of robot motion [grand challenges of robotics]," *Robotics & Automation Magazine, IEEE,* vol. 14, pp. 61-70, 2007.

[25] M. M. Quottrup, T. Bak, and R. Zamanabadi, "Multi-robot planning: A timed automata approach," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2004, pp. 4417-4422.

[26] M. J. Hornos and M. I. Capel, "On-the-fly model checking from interval logic specifications," *ACM SIGPLAN Notices,* vol. 37, pp. 108-119, 2002.

[27] G. Sirigineedi, A. Tsourdos, B. A. White, and R. Zbikowski, "Modelling and verification of multiple uav mission using smv," *Workshop on Formal Methods for Aerospace (FMA)* pp. 22–33, 2010.

[28] R. Simmons, C. Pecheur, and G. Srinivasan, "Towards automatic verification of autonomous systems," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2000, pp. 1410-1415.

[29] A. Medina Ayala, S. B. Andersson, and C. Belta, "Temporal logic control in dynamic environments with probabilistic satisfaction guarantees," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3108-3113.

[30] R. Muradore, D. Bresolin, L. Geretti, P. Fiorini, and T. Villa, "Robotic surgery," *Robotics & Automation Magazine, IEEE,* vol. 18, pp. 24-32, 2011.

[31] H. L. Younes, E. M. Clarke, and P. Zuliani, "Statistical verification of probabilistic properties with unbounded until," in *Formal Methods: Foundations and Applications*, ed: Springer, 2011, pp. 144-160.

[32] H. L. Younes and R. G. Simmons, "Probabilistic verification of discrete event systems using acceptance sampling," in *Computer Aided Verification*, 2002, pp. 223-235.

[33] H. L. Younes and R. G. Simmons, "Statistical probabilistic model checking with a focus on time-bounded properties," *Information and Computation,* vol. 204, pp. 1368-1409, 2006.

[34] D. C. MacKenzie, R. C. Arkin, and J. M. Cameron, "Multiagent mission specification and execution," in *Robot colonies*, ed: Springer, 1997, pp. 29-52.

[35] R. C. Arkin, D. Lyons, S. Jiang, P. Nirmal, and M. Zafar, "Getting it right the first time: predicted performance guarantees from the analysis of emergent behavior in autonomous and semi-autonomous systems," in *Proc. of SPIE Vol*, 2012, pp. 83871F-1.

[36] R. Jhala and R. Majumdar, "Software model checking," *ACM Computing Surveys (CSUR),* vol. 41, p. 21, 2009.

[37] E. Clarke, O. Grumberg, S. Jha, Y. Lu, and H. Veith, "Counterexample-guided abstraction refinement," in *Computer aided verification*, 2000, pp. 154-169.

[38] D. Dams, R. Gerth, and O. Grumberg, "Abstract interpretation of reactive systems," *ACM Transactions on Programming Languages and Systems (TOPLAS),* vol. 19, pp. 253-291, 1997.

[39] C. Böhm and G. Jacopini, "Flow diagrams, turing machines and languages with only two formation rules," *Communications of the ACM,* vol. 9, pp. 366-371, 1966.

[40] D. Lyons, R. Arkin, T.-M. Liu, S. Jiang, and P. Nirmal, "Verifying Performance for Autonomous Robot Missions with Uncertainty," in *IFAC Intelligent Vehicle Symposium*, Gold Coast, Australia, 2013.

[41] R. C. Arkin, *Behavior-based robotics*: MIT press, 1998.

[42] D. Lyons, R. Arkin, P. Nirmal, S. Jiang, T.-M. Liu, and J. Deeb, "Getting it Right the First time: Robot Mission Guarantees in the Presence of Uncertainty," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Tokyo, Japan, 2013 (to appear).

[43] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Y. Nakagawa, "Test arenas and performance metrics for urban search and rescue robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.