An Alert-Generation Framework for Human-Supervised, Multi-Agent Teams

Sarah Al-Hussaini[1], Jason M. Gregory[2], Satyandra K. Gupta[1]

***Abstract*— Multi-robot teams are useful in large-scale dangerous missions, and human-supervision is needed for critical decision making. Humans inherently lack in processing speed and power, and are prone to making mistakes, especially at stressful scenarios. We present an alert generation framework for humans to overcome these problems, and assign tasks better to improve resiliency. We demonstrate that our framework, based our state machine simulation and formal methods, can do probabilistic estimation on unfavorable events happening in the mission. We have introduced smart simulation for improving computationally efficiency compared to Monte-Carlo method. Moreover, for some cases, our inference-based method can provide guarantee on contingencies.**

I. Introduction

Multi-robot teams have great promise in dull, dirty, and dangerous applications, such as military applications and disaster-relief (DR) and humanitarian-assistance (HA) operations. Despite the ever-increasing capability of robotic systems, we believe human supervision will always be necessary because of diverse expertise, adaptive decision-making, the potential for synergy [1], and most importantly for critical decision-making in situations involving human lives and safety. HA/DR missions are extremely complex, and there is a large amount of uncertainty in the availability and efficiency of agents [2] due to the size and unstructured characteristics of the operational environment, limited communication, and the possibility of system and task-level failure. The dynamic conditions and intermittent data flow require the agents, including humans, to constantly adapt with the latest information. Human efficiency can decrease, especially in overly fatiguing and stressful situations of HA/DR missions. Humans are prone to making mistakes, and they have limited computational capability, and relatively large response time.

Alerts can help to prevent human-introduced inefficiencies and expedite decision-making, and thus improve resiliency. We already have alert systems deployed in a number of technologies for our every-day tasks, e.g., 'grammarly' software to prevent errors in our write-ups, lane departure warning and parking assistance systems to prevent vehicular accidents on roads. Likewise, an intelligent alert system can also provide great benefit to human-robot teams in safety-critical challenging applications. Researchers have already investigated the human factor concerns associated with supervisory control of multi-robot systems [3] and there exists several different alert-generating architectures and interfaces such as alert systems used by NASA [4]. There have also been several systems designed specifically for human-robot teams in the HA/DR context [5], which are purely reactive, and a notification is provided to the human once an undesirable event has occurred. A more relevant work is the predictive conventional interface and predictive virtual reality interface designs [6], where the risk and relevance of a robot performing some task are predicted. In this work, we also seek a proactive approach to providing alerts; however, our focus is on the generation of alerts for erroneous and inefficient, human-issued task assignments.

An alert system for multi-agent systems cannot simply rely on sensors and comprehensive observations unlike many alert systems available for our everyday tasks. It requires inference and probabilistic model estimation to account for the inherent uncertainty in mission operations. It also has to be flexible about the types of alerts offered, so that it can be tailored to the human preferences and mission needs. Thus, timely alerts can be generated in a meaningful way so that agents can take the necessary, corrective actions. In this work, we propose a novel, alert-generation framework that overcomes these challenges to improve resiliency of multi-agent teaming. We use probabilistic temporal logic to express human-specified alert conditions based on their preferences and requirements. We present an inference engine that compares the conditions with probabilistic outcomes from state machine-based simulations of the mission, and generate alerts as needed. We introduce smart simulation to provide computational efficient means to make good probabilistic estimates, and an inference-based technique to provide guarantees.

II. Problem Formulation

We consider a generic HA/DR mission in a very large environment, where a human-supervised team of robots is to be deployed. The robots may need to efficiently explore the affected regions, collect important information, and perform specified tasks. As the robot team navigates through the environment there is a nonzero probability of operational failure, which could be a result of spatial factors, system or subsystem failures or other stochastic events. Limited communication range (between two robots, and between humans and a robot), and the large scale nature of the operation cause huge delay in humans receiving information from the robots. We encode these typical challenges and characteristics of a generic HA/DR operation in our specific mission (described in Section III-A) in order to present and test our proposed alert generation framework.

Our framework takes the following as the inputs: (i) human-issued commands to the robots, (ii) estimated system

[1]Viterbi School of Engineering, University of Southern California, CA USA {salhussa, guptask}@usc.edu
[2]U.S. Army Research Laboratory, Adelphi, MD USA jason.m.gregory1.civ@mail.mil

models, and (iii) latest mission information. The framework also requires a list of mission-relevant alert conditions, specified by humans. Using the mission model, it determines whether a particular alert condition may hold $True$, given the most recent information. The goal is to estimate the likelihood of unfavorable events in a computationally efficient manner and generate alerts to the human supervisors.

III. Overview of Approach

A. Background

We assume, human supervisors provide high level instructions to each robot before deployment, and the robot carries out the operation. Then eventually, the robot returns to the humans to share information and receive new commands. Typical tasks for robots in a HA/DR mission include navigation, exploration, observation and information collection, identification of objects-of-interests, interact with environments/objects, collaborative tasks, which we abstract away using some basic tasks like *navigation, exploration, halting,* and *rendezvous*. For example, *rendezvous* typically stands for a pre-scheduled meeting between robots, however it can also be the first step of a pre-scheduled collaborative task. In order to improve resiliency in a mission where robotic failures are inevitable, robots need to attempt some *rescues* on temporarily-disabled robots to maximize performance. Any functional robot aware of a disabled robot can provide assistance to it for its revival, which we call a *rescue* attempt [2]. This rescue operation comes with its own risks, and it has stochastic outcomes. We also include *relay* task for robots which can be used to send one robot to another robot on field, and change the second robot's instruction-set. Since one of the focuses of this work is to provide alerts for future contingencies, this *relay* task can be particularly to useful for humans to execute new strategies in order to mitigate the adverse effects of potential contingencies indicated by generated alerts.

B. System Architecture

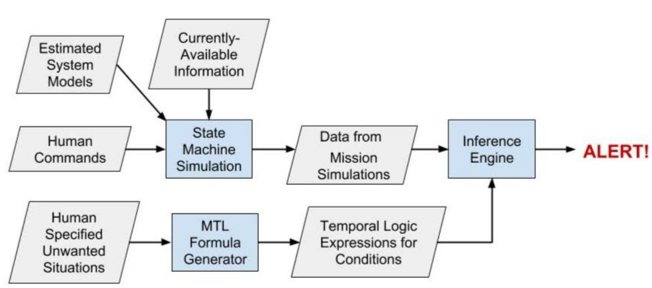


Fig. 1: Block diagram of the proposed alert-generation framework. The grey slanted rectacles are data blocks and blue rectangles are processing components of the system.

Since we are looking at large-scale challenging missions, the robots can be tasked for a long time, with a complex instruction-set with conditionals. First, The instruction-set given to each robot is converted to a task transition model as shown in Figure 2, and we use it to model each robot's behavior as a state machine. Whenever any robot returns to a human with new information, they provide state information of other robot teammates encountered in the field. We assume there are some estimated models of different stochastic parameters within the entire system. Using these models, the latest state information, and the state machine models of the robots, a series of simulations of the entire system are performed. Each simulation run is a collection of parallel, but inter-dependent, state machine simulations for all robots in the field. The results from simulations give probabilistic estimates on feasible outcomes in the mission. To do this, we perform simulation with intelligent sampling to minimize number of simulations needed to achieve a good probability estimation. Adaptive sampling is particularly important because the system may never be able to detect a low-probable critical situation with a limited number of simulations if simulations are done purely using Monte-Carlo method. Simultaneously, humans define their preferred list of unwanted situations that they feel are important to detect. These contingency conditions are then expressed as mathematical propositions. Our framework provides an inference engine that utilizes the results from the simulations, and finds *Truth* values for the user-specified alert conditions to issue alerts. This alert can be based on probabilistic estimates from the simulations (Section III-D), or it can provide guarantees on particular situations happening with 100% certainty (Section III-E). The proposed framework is represented in the block diagram in Figure 1.

C. Specification of Alert Conditions

We have identified some exemplary alert-triggering scenarios that may be useful or relevant, and have outlined the mathematical expressions for detection of these situations. We use different parameters and functions in order to formulate the conditions in a probabilistic temporal logic framework. Humans in a real-world mission might want different alerts for many new situations. Probabilistic temporal logic used here is a powerful language to mathematically express many kinds of complex conditions. Humans are free to choose different alert conditions from a potential list relevant to each mission, or craft their own conditions based on their preferences.

In our proposed framework, we use Metric Temporal Logic (MTL) specifications to detect certain characteristic within each mission simulation. Let Ψ be a set of atomic propositions, crafted from the aforementioned items relevant to the mission threads, and the MTL formulae are built from Ψ using Boolean connectives (and $\wedge$, or $\vee$, not $\neg$), propositions $\top(True)$ and $\bot$ $(False)$, and time-constrained or -unconstrained versions of temporal operators ($eventually\,\Diamond$, $always\,\Box$, $next\,\bigcirc$, $until$ $\mathcal{U}$). A time-constrained temporal operator is Γ_I, where $\Gamma \in \{\Diamond, \Box, \mathcal{U}\}$, and time interval $I \subseteq (0, \infty)$, while the unconstrained version is $\Gamma \equiv \Gamma_{(0,\infty)}$. $\mathbf{P}_{\sim p_{th}}\Phi$ indicates that the probability of Φ being $True$ is $\sim$ p_{th}, where, $\sim \in \{<, \leq, >, \geq, =\}$, $0 \leq p_{th} \leq 1$, and Φ is an MTL formula.

Example of some situations that human supervisors might

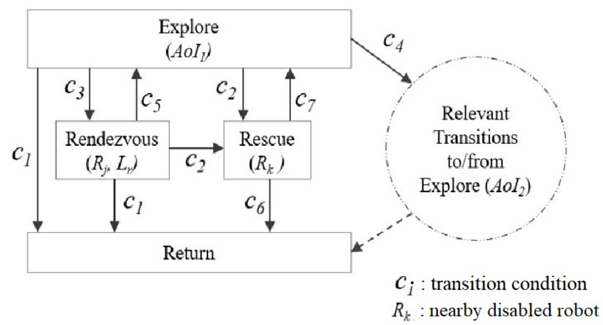
(a) Instruction in English

"Explore AoI_1, then explore AoI_2. Meet with robot R_j at location L_r between time T_1 and T_2.
If you see a disabled robot within d m distance, which has not been helped yet, attempt a rescue.
If the rescue is successful, resume exploration; if not, return home. At any cost return home before time T_f."

RobotInstruction 1

```
Explore AoI_1
Explore AoI_2
Rendezvous with Robot R_j between time T_1 and T_2 at location L_r
Return home before time T_f at any cost
while Exploring or navigating do
    if There is a disabled robot within d m distance which has not attempted for a rescue then
        Attempt rescue on it
        if The rescue attempt fails then
            Return home
        else
            Resume exploration
        end if
    end if
end while
```

(b) Pseudocode Generated from English Command



(c) Task Transition Diagram with Transition Conditions

Fig. 2: An example of the language decomposition offered by our proposed framework. English instructions provided by the human (a) are converted to pseudocode (b) that is used to generate a task transition model (c). Here, each AoI (*area-of-insterest*) is a specific region, and the rectangles in (c) are task names along with its arguments in the brackets.

want to get alerted about are (1) wasted time due to oscillatory states, (2) improbable rendezvous, (3) improbable or redundant rescue attempts, (4) navigation through risky region, (5) redundant exploration, (6) excessive travel time for rendezvous or rescue etc. In our work, we have specified condition corresponding to the detection of each above-mentioned situation, and its mathematical expression in MTL. Here, we are providing a simple case for improbable rendezvous where an alert is issued if there is a high probability ($\geq p_{th}$) of robot R_i and R_j never being together at the pre-scheduled rendezvous location X between time (t_1, t_2). MTL expression of this condition is $\mathbf{P}_{\geq p_{th}}[\Diamond_{(t_1,t_2)} \neg (L_{R_i} = X \wedge L_{R_j} = X))]$, where L_{R_k} is the instantaneous location of any robot R_k.

D. Using Simulation to Issue Alerts

We use discrete-time high level simulations of the mission to issue alerts. Each simulation uses the system model along with the latest information. It generates data on a single way the system can progress throughout the mission, out of infinitely many possibilities. We perform a number of simulations, and see what percentage of the simulations have a specific unwanted situation occurring. In stead of only relying on Monte Carlo runs, we do adaptive sampling. This means we simulate some missions with carefully-selected failure-modes (or contingencies) which can potentially influence occurrence of the adverse situation to be detected by the system. This can significantly reduces the number of simulations required to detect all possible adverse situations. Then we use the probability of those failure-modes (from estimated model of the system), along with estimated probability of certain alert condition given those failures occurring (from simulations with adaptive sampling), in order to get the overall estimated probability of certain alert condition to hold $True$. Human supervisors specify each condition they want to detect, along with a corresponding probability threshold. If the estimated probability meets the thresholding specification, the interface for humans shows an alert to notify them about the possibility of contingency.

The system model includes the robots' instructions along with the estimated models on stochastic parameters. Each task in the task transition model of a robot (Figure 2) is split into multiple states and their corresponding transitions. These individual state machines are dependent on each other and the environment. While staying within a state, a robot keeps performing some low-level actions, which affect itself (it's location, health, etc.), others, and the environment. At the start of the simulation, each state machine, i.e., robot model, needs to be properly initiated with appropriate state, history and other information corresponding to the latest update time for that robot. Then forward simulation is done on the entire system to assess probability of potential events in the past or future.

E. Using Inferences to Issue Alerts

The estimated probability of alert-triggering situations, calculated from simulations, might produce a poor representation of the real scenario. As the uncertainty and size of the system increase, the required number of simulations even with adaptive sampling can be very large. Also, it is difficult to build an adequate model of the stochastic parameters in the mission. Therefore, we attempt to perform quick inferences using only the non-stochastic parts of the system, (i.e., instruction-sets of the robots and maximum navigation speed) instead of using simulations to detect certain contingencies with significantly-higher confidence. For a complicated mission, it might not be always possible; nevertheless, it can produce useful results in some cases. Preliminary simulation-based results can aid in automated identification of the cases where this approach might be effective to provide guarantee.

Many of the alert conditions that we have mentioned refer to the probability of the system (or its agents) reaching a certain state within a time range. If an alert condition can be modeled in this way, and the simulation-based testing shows absolutely zero probability, this inference-based approach is attempted. This method firstly converts the state transition model of each robot into a directed graph without the transition conditions. Using a graph search algorithms to conduct reachability tests, and considering some simple Boolean literals, this approach can provide the humans with a much stronger assessment of an alert condition. In fact, in these cases it can guarantee absolute certainty. It may appear as though confirming reachability for an individual agent may be sufficient in this inference. However, HA/DR missions are actually much more challenging. For example, *rescue* and *relay* can complicate the inference process because they can make a disabled robot functional again and can change the state transition model of a robot by issuing a new instruction-set respectively.

Assume, we want to prove that robot R_i can not reach state S_f and location L_f by time T_f. In our research, we have found several conditions for providing guarantee for it. Here, we are providing two example conditions in the case of R_i being functioning; (1) S_f is not reachable from R_i's current state, and no functioning robot will attempt a rescue on any disabled robot, and not relay new instruction to R_i; (2) R_i's earliest possible arrival time (based on robot's maximum navigation speed and euclidean distances) to destination location L_f is later than time T_f.

IV. Preliminary Results and Conclusions

We have tested several alert conditions in our python-based custom simulator for some mission scenarios. We use estimated models for parameters related to failures, and high level task progress of the robots.

Let robot R_i be currently on the field, with the instruction-set described in Figure 2. For *improbable rendezvous* alert, let's assume we want to check the likelihood of event-A, which represents the case where "R_i does *not* reach rendezvous location L_r in time". There are several ways event-A may happen, but we only present selected cases as examples here. We carefully sample specific failure-modes which can affect the occurrence of event-A. As shown in the task transition diagram, *rescue* is a higher priority task, which can take over if there is a disabled robot nearby. Clearly, if a robot is found disabled near R_i's location during (or close to) its scheduled travel towards rendezvous, it can deviate for attempting a rescue, and eventually not have enough time to make it to rendezvous. For simplicity of the example, we are not describing other avenues like robotic failures from making rescue attempts. If the probability of robotic failure at a particular time instance is not very large, other robots' disability cases might not occur in a large fraction of simulations in Monte-Carlo method. Therefore, we do adaptive sampling where we specifically consider cases in which other robots get disabled at different locations and times (based on their models), and we estimate the probability of R_i missing rendezvous from simulations, given those failures. These conditional probabilities along with the probability models of robotic failures give the overall probability of event-A.

We also demonstrate that disability of a random robot can have impact on R_i's probability to rendezvous, and this effect is dependent on both the location and time components. If a disabled robot is never near R_i's location or it is found long before rendezvous or after rendezvous, this disability does not affect R_i's rendezvous. In our testing scenario we have found, if humans receive new information on one disabled robot in the environment, the estimated probability (from simulations) of event-A gets different values between $0-0.41$, depending on the combination of location and time. Realistically, this complex inferring can not be done easily or efficiently by human commanders themselves during a real operation.

Our alert generation framework shows promise in detecting different adverse scenarios and provide alerts. These alerts, when provided in a meaningful way, can assist in preventing human errors, and improving decision-making of humans supervising a multi-robot team in large challenging missions.

References

[1] A. H. DeCostanza, A. R. Marathe, A. Bohannon, A. W. Evans, E. T. Palazzolo, J. S. Metcalfe, and K. McDowell, "Enhancing human-agent teaming with individualized, adaptive technologies: A discussion of critical scientific questions," US Army Research Laboratory Aberdeen Proving Ground United States, Tech. Rep., 2018.

[2] S. Al-Hussaini, J. M. Gregory, and S. K. Gupta, "Generation of context-dependent policies for robot rescue decision-making in multi-robot teams," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spain, October 2018.

[3] S.-Y. Chien, Y.-L. Lin, P.-J. Lee, S. Han, M. Lewis, and K. Sycara, "Attention allocation for human multi-robot control: Cognitive analysis based on behavior data and hidden states," *International Journal of Human-Computer Studies*, vol. 117, pp. 30–44, 2018.

[4] B. K. Burian, J. A. Kochan, K. L. Mosier, and U. Fischer, "Autonomous, context-sensitive, task management systems and decision support tools ii: Contextual constraints and information sources," NASA/TM–2017–219539). Hampton, VA: NASA Langley Research Center, Tech. Rep., 2017.

[5] F. Jentsch, *Human-robot interactions in future military operations*. CRC Press, 2016.

[6] J. Roldán, E. Peña-Tapia, A. Martín-Barrio, M. Olivares-Méndez, J. Del Cerro, and A. Barrientos, "Multi-robot interfaces and operator situational awareness: Study of the impact of immersion and prediction," *Sensors*, vol. 17, no. 8, p. 1720, 2017.