Autonomous Search and Rescue with Modeling and Simulation and Metrics

Trevor J. Bihl
Air Force Research Laboratory, USA
Trevor.Bihl.2@us.af.mil

Chadwick Cox
Jacobs Corporation
Chadwick.Cox@jacobs.com

Yuki Adams
Applied Research Solutions
yadams@appliedres.com

James Pennington
Applied Research Solutions
jpennington@appliedres.com

Abstract

Unmanned Aerial Vehicles (UAVs) provide rapid exploration capabilities in search and rescue missions while accepting more risks than human operations. One limitation in that current UAVs are heavily manpower intensive and such manpower demands limit abilities to expand UAV use. In operation, manpower demands in UAVs range from determining tasks, selecting waypoints, manually controlling platforms and sensors, and tasks in between. Often, even a high level of autonomy is possible with human generated objectives and then autonomous resource allocation, routing, and planning. However, manually generating tasks and scenarios is still manpower intensive. To reduce manpower demands and move towards more autonomous operations, the authors develop an adaptive planning system that takes high level goals from a human operator and translates them into situationally relevant tasking. For expository simulation, the authors further describe constructing a scenario around the 2018 Hawaii Puna lava natural disaster.

1. Introduction

Unmanned aerial vehicles (UAVs) are seeing increasing use in location-based analysis, including search and rescue missions, due to their ability to physically cover more area than humans alone as well as their ability to operate in dangerous areas (Talamadupula, Schermerhorn, Benton, Kambhampati, & Scheutz, 2011). However, despite the operators being at a distance, manpower demands are pervasive in UAVs. While in layman's terms they might be considered as "autonomous" due to them not being directly controlled by a human operator (Talamadupula, Schermerhorn, Benton, Kambhampati, & Scheutz, 2011), in actuality there are still many functions that require human direction.

Planning is key to enabling autonomous systems whether they are biological or artificial intelligence (AI) agents (Russell & Norvig, 2009). In general, planning involves an interaction between a platform's actuators and its desired goals. This largely includes generating a detailed description of actions to be taken to accomplish a set of goals (Bihl, Cox, & Machin, 2019). In order to truly be autonomous, a system must further have the ability to monitor the execution of its plans, to re-plan when necessary, to determine appropriate goals, and to coordinate activities with other agents (Bihl, Cox, & Machin, 2019).

Sophisticated planners are available, but still require some degree of manual control, such as highly detailed pre-mission designated task list. Thus, although autonomous and automated operations of UAVs have been studied for decades, such operations are still manpower intensive and/or permit minimal replanning as situations change (Bihl, Cox, & Machin, 2019). Of interest are autonomous planning approaches that 1) resolve conflicting information, 2) plan, and 3) learn (Floreano & Wood, 2015).

Guiding this is understanding that true autonomy has three characteristics: 1) intelligent, informed, unforced choice, 2) an ability to handle uncertain and unexpected situations, and 3) a sense of self (Bihl, Cox, & Jenkins, 2018). These three features of autonomy are commonly identified across a wide set of diverse fields of study, including philosophy, psychology, law, government, robotics, cognitive science, and artificial intelligence (AI) (Bihl, Cox, & Jenkins, 2018).

In biological systems, these characteristics developed over a long time to address the richness of the world. To build synthetic automatons with these capabilities at realistic time scales, complex modeling and simulation (M&S) is required. A good M&S evaluation shows that an agent can make good choices, act robustly in the face of environmental variations, and successfully operate as a distinct entity within a group of cooperating and/or competing agents. Additionally, replicability challenges are a known limitation in location analysis research (Murray, 2021); however, repeatable and reproducible M&S scenarios and environments lend themselves to addressing replicability by providing a baseline mechanism.

Before deployment, an autonomous agent must be rigorously tested and evaluated to understand its

URI: https://hdl.handle.net/10125/102853
978-0-9981331-6-4

decision spaces and to develop expectations of performance (Bihl & Talbert, 2020). Though there has been some discussion of explainable AI, the complexity of the decisions being made by an autonomous agent and the tempo of its mission might preclude understandable explanations in some situations. Thus, M&S for developing trained and trusted autonomous agents is of interest herein; this can be thought of in much the same way as service dogs that, though trained, tested, and trusted, are not query-able (Bihl & Talbert, 2020). However, the research herein focuses on symbolic methods that might provide the data that are needed to construct explanations. Together, it is hoped that some combination of explanation and testing will be sufficient for certification.

In this paper, we study dynamic and autonomous goal/task flexible operations of UAVs as environments are traversed and declarative data are gathered. Results from sensing operations are considered by a probabilistic decision agent to determine the course of action multiple UAVs should pursue. In the proposed *Adaptive Planning* methodology, all UAV actions are autonomously selected by a central controller, which is the primary UAV in the swarm. Human operator input is not necessary beyond the initialization of the mission with the selection of the goal condition.

2. Background

2.1. UAVs in Search and Rescue (SAR)

Traditional research on AI/ML for search and rescue involves sensor-centric approaches to finding objects of interest. This includes schduling, planning, and tasking of sensors and their associated platforms (Hero & Cochran, 2011) (Musick & Malhotra, 1994). But this largely involves detecting an object and then cuing a human operator for next actions. Herein, of interest are fully autonomous search and rescue operations whereby an autonomous UAV would detect, identify, and then prescribe an appropriate course of action. Prior work, e.g. (Scherer, et al., 2015), developed autonomous SAR whereby UAVs reacted to sensed data with onboard planning, control, and sensor data exploitation; however, the system in (Scherer, et al., 2015) was not responsive to autonomously generating tasks. More recently, research has developed abilities for decentralized tasking of multiple assets in the presense of uncertainty (Liu, Seo, Yan, & Tsourdos, 2020); however, this work did not consider core abilities to self task and generate new tasks, which will be key to future AI operations.

2.2. Artificial Intelligence

AI involves a complex interaction of algorithms, software, hardware, applications, and data. Finding the correct algorithms in the proper combinations is not as easy as some early researchers expected. Because of the generally reprogrammable nature of the underlying computer hardware, the space of possible algorithms is immense. In response to this difficulty, AI research has focused largely on methods that are mathematical, statistical, and rule-based in nature, c.f. (Russell & Norvig, 2016) (Duda, Hart, & Stork, 2012), to quickly address narrow, but very useful applications. AI can be grouped in a rough taxonomy by the nature of each innovation. Broadly, AI research is either application based, where a known algorithm is applied to an application, or theory based, where researchers develop, characterize, or expand algorithms to address classes of computational problems (Silver). Areas of theory based AI approaches include those that develop attributes largely viewed as necessary for intelligent behavior (Russell & Norvig, 2016) (Luger & Stubblefield, 2004) (Nilsson, 1998) (Poole, Mackworth, & Goebel, 1998) (Håkansson & Hartung, 2020). These areas include: reasoning, knowledge representation, planning, learning, human-computer interaction, and integration. These areas include further overlap with other domains, i.e. human-computer interaction overlaps with robotics and reasoning overlaps with neuroscience and cognitive architectures (Zacharias, 2019).

2.3. Automation vs Autonomy

Related to true AI are autonomous capabilities. To understand what is meant by autonomy, we must understand the current state of the art for machine intelligence and how it relates to automation and autonomy. For this purpose, we will consider the following definitions, adapted from (Bihl & Talbert, 2020):

- **Automation** is where a system functions with little to no human involvement, but with well-defined tasks with predetermined outcomes.
- **Autonomy** is where a system has intelligence-based capabilities, allowing it to respond to unexpected and unanticipated situations.

Central to these distinctions is that an autonomous system can select an appropriate task or goal to pursue, modify its thinking constructs, and appropriately assume roles (Bihl, Cox, & Jenkins, 2018). Selecting the appropriate task or goal to pursue further implies reasoning, including planning capabilities that leverage models of the self and of the environment.

2.4. Planning

Planning is a pervasive problem for robotics and for UAVs, it is also central to many autonomous capabitlities. Planning involves a hierarchy of terms and functions which range from the highest level of mission planning down-to the lowest level of determining forces to apply to specific vehicle actuators (Bihl, Cox, & Machin, 2019). The general conceptualization of planning as a protocol stack is presented in Figure 1. Here, we present planning as a hierarchical relationship between an operator/user and the actuators. This encompasses the breadth of planning, e.g. in robotics planning describes control of motion (Ghandi & Masehian, 2015), whereas in artificial intelligence planning is more abstract and implies a set of tasks or actions (García-Martínez & Borrajo, 1997). When a plan is executed, the proper sequencing of actions are expected to make the agent reach a goal, usually by improving some value function of the problem state. At each increasing-level of the planning stack, a planner operates on an increasingly abstract notion of state to achieve an increasingly general objective. Overall, plans and planning can be thought of per the following definitions, from (Bihl, Cox, & Machin, 2019):

- A **plan** is defined as a detailed description of actions to be taken by one or more entities to accomplish a set of goals.
- **Planning** means to generate the plan. This generation is subject to a set of constraints that limit the plausible choices of actions.

For the purposes of this research, planning is considered as occurring between the operator and the vehicle's autopilot (Figure 1). Of particular interest are mission and task planners that are given abstract goals. The planners must determine the tasks, schedule of tasks, and allocate resources to achieve these goals. In operation, as conceptualized in Figure 1, a mission planner selects tasks, determines the schedule, and then employ tasks planners to complete the plan.

At a high level, a mission is a set of tasks (Botelho & Alami, 1999). As described in (Bihl, Cox, & Machin, 2019), each task can be viewed as a tree structure that decomposes the task into subsequently finer levels of detail. The leaves of the tree are basic actions that are directly executed by vehicles, sensors, and other assets. The required actions usually include driving the vehicle, so a path planner generates waypoint paths. Planning also involves some scheduling to ensure that tasks are coordinated and accomplished at the appropriate times. It also includes asset allocation to assign vehicles, weapons, sensors, and other resources to each task. Mission planning can become a complicated set of intertwined sub-planning efforts. Iteration is often required to resolve the interdependency of the various planning functions.

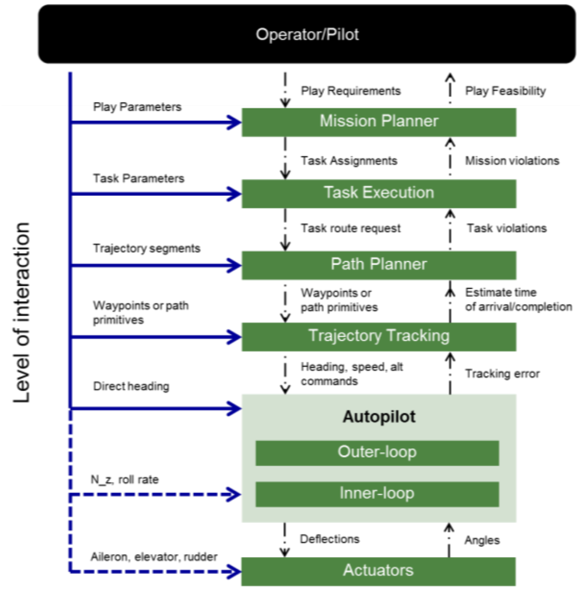

Figure 1. The planning stack, from (Kingston, 2017).

When the events of the real world do not match the expectations of the planner, changes to the plan are often needed. The solutions to this problem is known as plan revision (Williams & Burdick, 2006) and replanning (Tate, 1990). Plan revision attempts to address a discrepancy with minimal modifications to the existing plan. Replanning involves major modifications and may even begin with a clean slate, including new objectives. Small discrepancies, e.g. unexpected obstacles, in expectations are handled by revision planning at the lower-levels. Large changes to the environment, such as a missing target, may require task changes. Very small unexpected events, such as a wind gust, might not need replanning at all if the autopilot is sufficiently robust. However, accumulated, small, unexpected events, such as frequent wind gusts, may impact resources, such as fuel, and therefore require task replanning.

3. Adaptive Planning for Autonomy

An autonomous artificial agent is expected to make proper choices within uncertain and unexpected situations in a flexible manner. It is important for the agent to select appropriate actions, to modify its thinking constructs, and to appropriately assume roles (Bihl, Cox, & Jenkins, 2018). For UAV applications, these high-level, cognitive functions are supported by asset allocation, scheduling, trajectory generation,

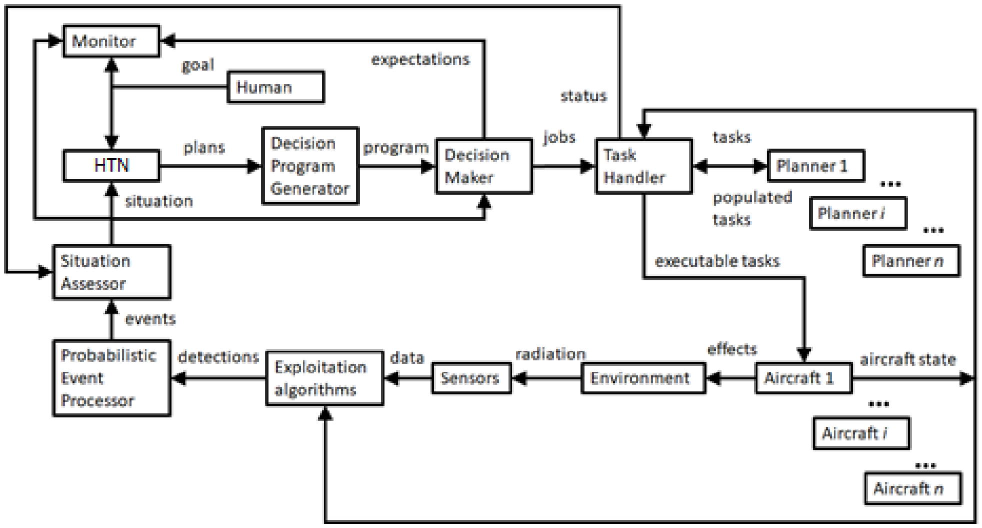

Figure 2. Conceptualization of an autonomous planning and tasking process

flight control, obstacle avoidance, and sensor data exploitation. While still considered as planning, such low-level functions have been automated for decades (Floreano & Wood, 2015).

3.1. Decision Making Space

The developed autonomy component, Figure 2, prosecutes a mission by choosing situation-appropriate tasks in response to events. The key components of this system are a status monitor, a Hierarchical Task Planner (HTN) (Gorgievski & Aiello, 2014), a decision maker, low-level planners, aircraft, sensors, a probabilistic event processor, an a situation assessor. In the top-left corner of Figure 2 is the automated monitor that is aware of the goal of a human operator. The monitor watches the evolving situation and also considers intermediate expectations that are generated by the decision maker. The monitor may trigger the HTN to replan if expectations are not being met. The HTN generates multiple plans for achieving the goal from the initial state. These plans are used to generate a probabilistic decision program. This decision program triggers tasks, based upon the present situation. The program consists of a set of probabilistic rules that map states to appropriate tasks, as well as expected postconditions. Decision rules are prioritized using value information that is determined by the HTN. The tasks are passed from the decision maker to a task handler. This handler passes individual tasks to appropriate low-level planners for asset selection, scheduling, and route generation. The best planning service is selected for each particular task. Fully populated tasks are returned to the task handler. Each task is passed to the aircraft guidance systems. As the aircraft move, they detect objects with their sensors. These detections are probabilistic and are passed to a probabilistic event handler that evaluates each relevant event. Event probabilities are passed to the situation assessor with status information to maintain a state vector.

3.1.1. Hierarchical Task Network (HTN) Planner. The high-level planner, SHOP++, is an extension of the Python version of the Simple Hierarchical Ordered Planner (SHOP) (Nau, et al., 2005) (Nau, 2013). SHOP++ is contingency aware, meaning that it understands that tasks may not achieve their expected postcondition. Therefore, it generates many plans. SHOP++ allows a user to describe primitive tasks in terms of transitions from preconditions to postconditions. Each precondition describes a state of some system to which the task is applicable; further any state representation is supportable and virtually any transformation function may be implemented. Each postcondition describes how a task can transform that state. The HTN further uses a double-ended queue to perform a tree search for solutions as viable sequences of tasks that move initial conditions to a goal condition. Like other HTNs, *methods* allow a planner to achieve in a single iteration what would ordinarily require a search of many branches. Methods are collections of primitive tasks and are built during planning.

The HTN is also contingent and the search will find multiple possible solutions given multiple possible postconditions. A set of two or more postconditions represents the various ways that a task may transform the state. The first positcondition is expected when the task is executed properly. The other postconditions represent non-ideal states that might occur if something goes wrong. These possibilities are prioritized and probability values may be provided if they are available. The search will find additional paths that include the off-nominal conditions, allowing the system to handle these contingencies. The priorities and probabilities are used to modify the values of tasks, thereby ordering the deterministic search without explicitly resorting to probabilistic search methods. In this way, the most important contingencies are considered first in case the planner hits a time limit.

3.1.2. Decision Maker. The purpose of the decision maker is to quickly drive a succinct action policy, by issuing task commands for subsequent execution as actions. The decision maker avoids complicated replanning within the agent's primary loop. Fast execution is necessary for timely responses to quickly changing conditions.

The process is analogous to a sports game, where a coach deliberates and builds a succinct play from a set of template actions. The play is then executed in real time by the players. Using a sports analogy, in this system, SHOP++ is the coach and the decision maker and downstream components, including the task hander and the low-level planners, stand in the place of the players. The decision program that comes from SHOP++ via the program generator is the play.

The decision maker is an automation routine that monitors the situation in real time and selects any task that is appropriate for the conditions at hand. It is similar to a rule-based system, but it evaluates conditions with Python program strings, not just comparison operators or distance measures. The decision maker is probabilistic so it handles an uncertain assessment of the situation. The decision maker annotates the task with information that informs the task handler about details, such as the area where the task is to be applied. The decision program returns a nominal expectation, which is the most probable postcondition that will arise if the task is successfully executed. This postcondition is used by the monitor to make sure that the task is successfully executed. The decision program runs in real time, making the agent much faster than if SHOP++ is inside the decision loop.

3.1.3. Decision Program Generator. The Decision Program Generator (DPG) converts a set of plans from SHOP++ into a computer program (Python herein) for the decision maker. Tasks within the SHOP++ plans become the tasks of the decision making program. The preconditions of the SHOP++ tasks become the triggers for the decision program's tasks. The postconditions become the expectations that are produced by the decision program. The plan values that are determined by SHOP++ are used to prioritize tasks. The result, in a perfectly deterministic world, is the equivalent of a state machine that executes the plans. If all the preconditions transform to the expected postconditions then the nominal plan will be executed. If the state transforms in an unexpected fashion then contingent tasks are available. In a probabilistic world, the decision program exhibits considerable flexibility over a state machine because it can operate even in the presence of unexpected state transitions.

3.1.4. Probabilistic Event Processor. The Probabilistic Event Process (PEP) presently accepts probabilistic classifications from one or more sensor exploitation algorithms. For example, one can define an event as "(two or more sedans and one or more trucks) or (no vans and between three and four motorcycles)". The algorithm accumulates probabilities of possible combinations of object counts, making it potentially computationally complex. However, there are a features that speed execution greatly. First, the PEP assumes that detections are independent and a closed-world assumption is included, reducing the possible combinations of objects that must be considered. (Conditional probabilities could be considered if they were available.) Second, the software orders the combinations according to decreasing probability, so that results may be approximated by truncating the process after accumulating only a small subset of the total collection of combinations.

A general set of predicates is planned for the PEP. These predicates will operate together with the object counts to provide more general descriptions of events. In additions, the PEP will be recursive, providing hierarchical descriptions as events of events.

4. Analysis and Simulation

In order to assess the autonomy of the developed framework, an appropriate M&S scenario, or challenge, is needed. Ideally, such scenarios have richness and complexity that preclude solutions with simpler systems.

4.1. Search and Rescue Richness and Complexity and Performance Metrics

During search and rescue missions, autonomous agents could conceivably accept dual tasks: primarily searching for cars and fiducials/hazards, as well as rescuing cars by dissuading them from traveling on dangerous roads. Dissuading means that a fiducial is detected and that cars heading towards it are stopped. For illustrative purposes, we will assume that an agent can immediately apply all its assets to dissuade known cars from known hazards. However, the situation is dynamic and cars can enter and exit the scenario as hazards expand, start, or contract. Therefore, an agent must use some of its assets to search the environment to keep its assessment of the situation up to date. To assess how well the agent handles these dual tasks, two metrics were constructed. These include, the vehicle detection efficiency:

$$\varepsilon_{VD} = N_{VD} / N_V \tag{1}$$

where N_{VD} is the number of vehicles detected and N_V is the number of vehicles. The fiducial detection efficiency:

$$\varepsilon_{F} = N_{FD} / N_F \tag{2}$$

where N_{FD} is the number of fiducials detected and N_F is the number of fiducials.

Since real world missions do not often have a specific ending, as the truth is never fully known during the mission, neither should sufficiently complex simulations. Thus, metrics that measure the goodness of the state at the end of the mission or that measure the total time of the mission have no meaning. The benefit of metrics (1)-(3) is that, by recomputing at intervals and averaged over a long period of time, we can assess how well an agent is performing the exploration task. The metrics (1) and (2) additionally have the benefit of being bounded between 0 and 1.

4.2. Real World Context: 2018 Puna Eruption

Inspiration for a rich and complex scenario was taken from the 2018 volcano eruptions on the Big Island of Hawaii. Beginning on May 3, 2018, earthquakes and spewing lava began disrupting regular life in lower Puna on the Big Island of Hawai'i (Wikipedia, 2020) (Overview of Kīlauea Volcano's 2018 Lower East Rift Zone Eruption and Summit Collapse, 2019). Figure 3 presents a general map of the Big Island of Hawaii, along with the 9 administrative districts on the island. Lower Puna encompasses the part of District 1 that includes Kapoho and the Leilani Estates. For context, Also labeled are the 5 volcanos of the Big Island of Hawaii (Kilauea, Mauna Loa, Mauna Kea, Kohala, and Hualalai). Kilauea was the volcano associated with the 2018 lower Puna eruption. This eruption notably occurred in the Leilani Estates subdivision and the community of Kapoho, with the result being the destruction of many homes and farms. If UAVs had been available for search and rescue operations in the 2018 eruption, they might have saved lives by performing general tasks such as reconnaissance, dissuasion, and rescue.

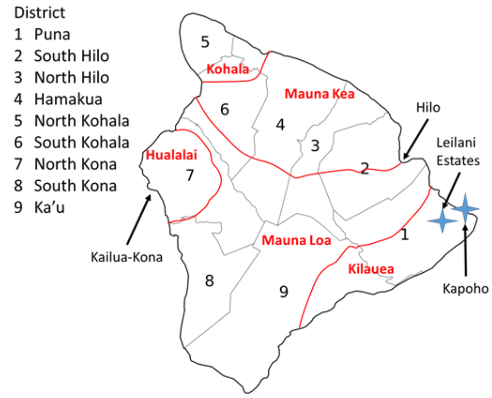

Figure 3. Big Island of Hawai'i with districts (1-9) identified along with the five volcanos and two points of reference (Kailua-Kona and Hilo). Lower Puna encompasses the part of District 1 that includes Kapoho and the Leilani Estates.

Figure 4. Google Map of Lower Puna eruption, as of September 18, 2018 from (A Community Lava Map Project, 2018). This map includes Kapoho (far right, covered in lava as indicated by pink shaded areas) and the Leilani Estates (green/yellow street grid in middle of figure). Icons represent hazards (gas, lava,etc.)

Overall, the eruption spanned May 3 September 4 and involved 13.7 square miles of land being covered with lava, destroying 700+ homes, and 1.36 square miles of new land being created in the ocean (Wikipedia, 2020). This is encapsulated in the Geographic Information System (GIS) representation of the lava flows and events in Figure 4. Figure 4 is from September 18, 2018 and shows the lava flow at its greatest and final extent. In Figure 4 we see various warnings, road conditions, lava coverage, as well as a general street map. Colors in Figure 4 indicate conditions, with Light Pink indicating older inactive flows, Green roads indicating passable roads, yellow roads indicating authorized personnel, and symbols indicating various conditions (volcano symbols for eruptions, triangles for hazards, fires, etc.).

If UAVs had been available for search and rescue operations in the 2018 eruption, we can envision them in general tasks of interest to exploring how heterogeneous planners can operate together for the same mission, an appropriately rich and complex scenario was developed.

4.3. Simulation Scenario Development

The GIS representation in Figure 4 was used to create a scenario. SLAMEM, a simulation environment built by Toyon Research Corporation, was used as the modeling environment due to its primary focus on simulating intelligence, surveillance, and reconnaissance (ISR) missions for UAVs (Sullivan, Agate, & Beckman, 2004). Thus, tracking, vehicle movement, identification, and ISR missions from UAVs are standard tasks for SLAMEM.

The process followed to move from the GIS representation in Google Maps to a UAV relevant in the SLAMEM simulation environment are as follows:

1. The overall image from the location was taken, bounding box of lat/long of the extreme points
2. The M&S system (SLAMEM) took the lat/long and found its internal map representation of this area
3. Road networks were extracted manually
4. Polygons to represent and illustrate location areas of lava on the map were added
5. Lava fiducials were added where the lava intersected the roads
6. Lava fiducials had logic whereby they appear over time, simulating lava flowing out
7. Vehicles were added to the road as well.

Alternatives to this could be extracting *.kml files to the desired simulation environment and adding sprites to represent vehicles and lava fiducials.

4.4. Simulation Scenario

In Figures 5a and 5b, the lava (orange), is seen to cross several roads (purple) and represents hazards to vehicles (oranges X's). UAV search routes are then represented in yellow. Figure 5a presents the scenario evaluated using the *baseline* non-autonomous exhaustive search of lawnmower patterns; Figure 5b presents the presents the scenario evaluated using the *adaptive planning* approach where paths and tasks are learned in-mission as new data is discovered.

In both the *baseline* and *adaptive planning* cases, 5 UAVs start at the upper left (notionally a staging area off the road from Pahoa to Hilo) and the scenarios all started with 32 cars, and an eventual 14 lava hazards (which grow dynamically over time). The lava hazards are not identical to Figure 4, but similar in scope. Notably, there are more vehicles than hazards, making monitoring difficult, but room remains for ground vehicles to maneuver significantly, making the vehicles harder to track. Additionally, the number of these hazards slowly increases as the lava moves. The simulation does not provide any prior knowledge of these hazards to the autonomous system and it must find the hazards by searching for them.

Vehicles were modeled as simple automatons for simulation, driving randomly across the road network. At an intersection, a driver will continue on the present road or to take one of the available turns. Without a road advisory, a ground vehicle that is driven towards a lava hazard is 50% likely to cross it and be destroyed. With a hazard advisory, the ground vehicle is only 20% likely to be destroyed because it is aware of the hazard. If an air vehicle is attempting to dissuade a ground vehicle from approaching a hazard, the ground vehicle is only 5% likely to strike the hazard and be destroyed. Once a driver encounters and survives a particular hazard, she will forever avoid that hazard.

In operation, UAV detections occur when an object falls within the footprint, the polygon that is formed by the intersection of a sensor's field of view and the ground. The UAVs have the responsibility to find lava

Figure 5. Graphical representation mission progress with UAV tasks/paths for same mission with no autonomy (left) and with the proposal (right)

hazards on roads and then issue an advisory for that road segment, dissuading vehicles from driving on it. They also look for vehicles to save before they hit a lava hazard. Additionally, the UAVs must search the road network to maintain an up-to-date understanding of the situation. However, they must occasionally break from this search, exploiting their knowledge to save ground vehicles. Thus, the problem is one of exploration versus exploitation.

specified because of a closed-world assumption. The agent will begin with searches of the road segments.

The nominal (lowest cost) plan that the HTN discovers is *Search*(R_X) for all R_X and then *Warn*(R_X)) for all R_X , terminating in the precondition of Complete, which is the goal. This plan handles the case where all cars follow the advisories. The longest plan that the HTN builds is *Search*(R_X) for each road then *Warn*(R_X) for each road then apply *Dissuade*(R_X) to all roads, resulting in the goal. This plan handles the case where no car follows the advisories. While this example is relatively simple, the HTN could have just as well found these plans if we had included thousands of irrelevant tasks and thousands of irrelevant situation variables. Thus, complexity reduction is an important function of the HTN which discovers the relevant tasks to put into the decision program.

Table 1. Tasks of the High-level Planner

Task	*Precondition*	Postconditions	Description
Warn(R_X)	lava_on(R_X) and not(warn_cars_on(R_X))	warn_cars_on(R_X) and not(car_on(R_X))	Lava has crossed this road and there has been no advisory issued for this road so issue and advisory. Cars will either obey this advisory, or not.
		warn_cars_on(R_X) and car_on(R_X)	
Search(R_X)	not(certain_of(R_X))	not (lava_on(R_X)) and not(car_on(RX)) and certain_of(R_X)	If the agent is not certain about a road then it will search it. The result is certainty of some combination of lava danger and cars being on the road or not on the road. After an aircraft completes a search, the certainty is set to 1.0 and then slowly declines.
		not (lava_on(R_X)) and car_on(R_X) and certain_of(R_X)	
		lava_on(R_X) and not(car_on(R_X)) and certain_of(R_X)	
		lava_on(RX) and car_on(R_X) and certain_of(R_X)	
Dissuade(R_X)	warn_cars_on(R_X), car_on(R_X)	not (car_on(R_X))	If an advisory has been issued for a road and a car is on it anyway, apply a certain dissuasion tactic. To ensure the algorithm terminates, the car is assumed to leave the road segment.
Complete	((not(car_on(R_X)) or not(lava_on(R_X))) and certain_of(R_X))) for all R_X	N/A	A plan is complete if it achieves all cars off of all roads that do not have lava dangers, with certainty.

The automation for the Hawaii scenario is described by three primitive tasks, Table 1, with each of these tasks having a single preconditions and one, two, or four postconditions. Additionally, there is a completion task. This completion task will never be executed by the decision maker, because the absolute certainty requirement will never be received, but some condition is necessary for the HTN plans to terminate. Each task and each condition variable has an instance for each road segment, $R_{X,x}$. The use of the road segments provide an example of how general knowledge can be instantiated by relating tasks and variables to members of object sets. Because of the postconditions and the road segments, the number of unique tasks are the permutations of seven times fifteen plus one. The primary contingency here is that if a car does not obey the warning then the agent will perform a more significant dissuasion. The starting condition is *warn_cars_on*(R_X) and *not(certain(*R_X*))* for all R_X, meaning that no advisories have been issued and the agent has no knowledge of what is on any road. The other variables are assumed to be *False* until otherwise

4.4. Simulation Results

Figure 6a shows a plot of the three performance measures for both non-autonomous (exhaustive lawnmower pattern searches) and autonomously directed searches. Figure 6b shows the raw values of cars detected, lava detected for the same two cases. By the end of the simulation at 2,695 seconds, with the *Baseline*, 20% of the 32 ground vehicles survive and less than 80% of the vehicles are found. In contrast, with the *Adaptive Planning* system, 55% of the ground vehicles survive and 100% of the vehicles are found. The reason that not all of the vehicles survive, even if found, is due to the simulation allowing vehicles to continue to move after a dissuasion is completed and thus they could hit a different lava hazard.

Notably, some behaviors are evident in the results seen in Figure 6a/b. In simulation, each air vehicle follows one road segment at a time, with its camera oriented ahead in a push-broom fashion. Without missing any portion of a present road segment, each aircraft occasionally points it sensor to nearby road segments, which is associated with greatly increasing search efficiency and jumps in detection efficiencies.

planning while potentially considering many task elements, situation variables, and goal conditions. Because this planning can be too complicated to run in real time, the resultant plans are converted into a decision program.

A complex scenario is used to demonstrate the autonomy system. This scenario is based on the 2018 volcano eruption on the island of Hawaii. When using the *adaptive planning* approach, the system of agents is shown to manage exploration and exploitation while

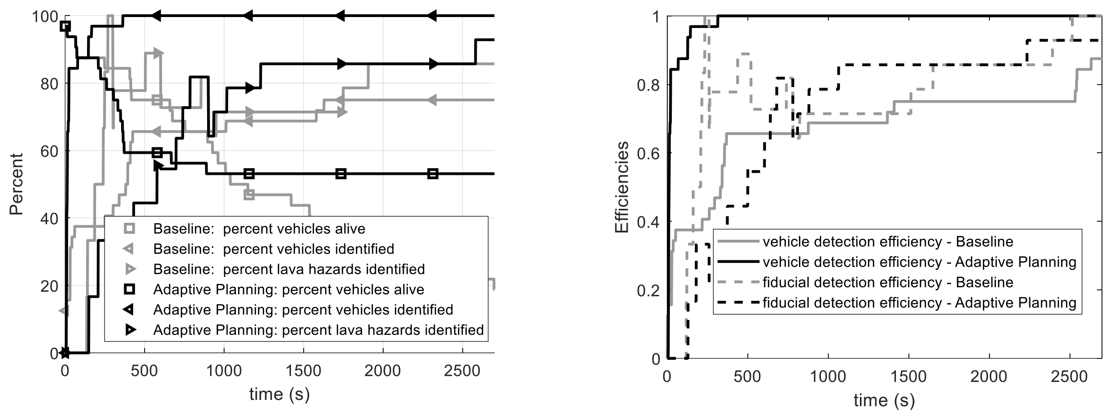

Figure 6. Raw results (a, left) and results relative to metrics (b, right)

Fiducial detection efficiencies, when detecting the lave, is mostly increasing too and sees occasional drop which occurs when new lava crosses a road. Ground vehicles are found at a much lower rate and with much lower confidence because they are swiftly moving targets, they are small, and they are often seen at long distances. So, very few dissuasion actions are performed by the automaton. The most obvious solution to this problem is to apply more air vehicles across this very large area.

5. Conclusions

A new, *adaptive planning*, approach to autonomy is presented to achieve useful behaviors within a complex environment. The automaton applies a high-level planner to select the correct tasks elements for the conditions and the goal at hand. As a first step towards, handling unexpected situations, the planner considers contingent state transitions. The plans are used to generate a decision agent that intelligently chooses appropriate actions in response to changing conditions.

The agent addresses the complexity of the environment by separating modes of "thought" across two time scales. The planner provides deliberation on a slow time scale where it can perform complex directing multiple aircraft during a search and rescue mission. Notably, the *adaptive planning* system reduces the danger to ground vehicles faster than when using the *baseline* nonautonomous operation.

Improvements are planned. All the components will benefit from learning. Future research will also give the agent a sense of self with access to its own internal states, including the states of its AI algorithms. A realism that is not included in this simulation is that each driver should have a specific destination and only drive for a limited time. Additionally, adding in probability of a driver to respond to an autonomous dissuasion was not considered, but would add more realism since some drivers would ignore such suggestions. Cars tend to survive much longer when aid is provided by the autonomy system, which means that most cars with destinations would remain safe. Finally, adding stochastic vehicles/fiducials as well as stochasticity of the autonomous agents and their success would further add realism.

6. Acknowledgements

The views expressed in this paper are those of the authors and do not represent the views of any part of

the US Government. This work was cleared for unlimited release under: AFRL-2021-2135.

7. References

A Community Lava Map Project. (2018, Sept. 18). *2018 Hawai'i Island Eruptions*. (hawaiilavamaps@gmail.com, Editor) Retrieved Feb. 5, 2020, from Google Maps: https://www.google.com/maps/d/u/0/viewer?hl=en&mid=1CvBhH9wEeztBrqYbsGDi4YjU1k1QH5AL&ll=19.484513950639798%2C-154.88181773919212&z=13

Bihl, T. J., Cox, C., & Machin, T. (2019). Towards a Taxonomy of Planning for Autonomous Systems. *IEEE NAECON*.

Bihl, T., & Talbert, M. (2020). Analytics for Autonomous C4ISR within e-Government: a Research Agenda. *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2218-2227.

Bihl, T., Cox, C., & Jenkins, T. (2018). Finding common ground by unifying autonomy indices to understand needed capabilities. *SPIE Defense and Commercial Sensing*.

Botelho, S., & Alami, R. (1999). M+: a scheme for multi-robot cooperation through negotiated task allocation and achievement. *IEEE International Conference on Robotics and Automation*, 1234-1239.

Duda, R., Hart, P., & Stork, D. (2012). *Pattern classification*. John Wiley & Sons.

Floreano, D., & Wood, R. (2015). Science, technology and the future of small autonomous drones. *Nature, 521*(7553), 460-466.

García-Martínez, R., & Borrajo, D. (1997). Planning, learning, and executing in autonomous systems. *European Conference on Planning*, 208-220.

Ghandi, S., & Masehian, E. (2015). Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design, 67*, 58-86.

Gorgievski, I., & Aiello, M. (2014). Georgievski, I., & Aiello, M. (2014). An Overview of Hierarchical Task Network Planning. *arXiv prepublication*. Retrieved from https://arxiv.org/abs/1403.7426

Håkansson, A., & Hartung, R. L. (2020). *Artificial Intelligence : concepts, areas, techniques and applications.* Studentlitteratur AB.

Hero, A., & Cochran, D. (2011). Sensor management: Past, present, and future. . *IEEE Sensors Journal, 11*(12), 3064-3075.

Kingston, D. (2017, Dec. 7). *UxAS Overview.* Retrieved from https://www.youtube.com/watch?v=B6xIIcAoiwU

Liu, R., Seo, M., Yan, B., & Tsourdos, A. (2020). Decentralized task allocation for multiple UAVs with task execution uncertainties. *International Conference on Unmanned Aircraft Systems (ICUAS)*, 271-278.

Luger, G., & Stubblefield, W. (2004). *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* (5th ed.). The Benjamin/Cummings Publishing Company, Inc.

Murray, A. (2021). Replicability Challenges in Location Analytics. *Hawaii International Conference on System Sciences (HICSS)*, 5367-5375.

Musick, S., & Malhotra, R. (1994). Chasing the elusive sensor manager. *Proceedings of National Aerospace and Electronics Conference (NAECON'94)* , 606-613.

Nau, D. (2013). *PySHOP Documentation and Code.* University of Maryland. Retrieved from https://bitbucket.org/dananau/pyhop/src/default/README.md

Nau, D., Au, T., Ilghami, O., Kuter, U., Wu, D., Yaman, F., . . . Murdock, J. (2005). Applications of SHOP and SHOP2. *IEEE Intelligent Systems, 20*(2), 34-41.

Nilsson, N. J. (1998). *Artificial Intelligence: A New Synthesis.* Morgan Kaufmann.

(2019). *Overview of Kīlauea Volcano's 2018 Lower East Rift Zone Eruption and Summit Collapse.* United States geological Survey. Retrieved from https://volcanoes.usgs.gov/vsc/file_mngr/file-224/OVERVIEW_Kil2018_LERZ-Summit_June%202019.pdf

Poole, D., Mackworth, A., & Goebel, R. (1998). *Computational Intelligence a Logical Approach.* New York: Oxford University Press.

Russell, S., & Norvig, P. (2009). *Artificial intelligence: a modern approach.* Prentice Hall.

Russell, S., & Norvig, P. (2016). *Artificial intelligence: a modern approach.* Pearson Education Limited.

Scherer, J., Yahyanejad, S., Hayat, S., Yanmaz, E., Andre, T., Khan, A., . . . Rinner, B. (2015). An autonomous multi-UAV system for search and rescue. *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, 33-38.

Silver, T. (n.d.). *Lessons from My First Two Years of AI Research.* Retrieved Aug. 5, 2019, from https://web.mit.edu/tslvr/www/lessons_two_years.html

Sullivan, K., Agate, C., & Beckman, D. (2004). Feature-aided tracking of ground targets using a class-independent approach. *SPIE Proceedings*, 54-65.

Talamadupula, K., Schermerhorn, P., Benton, J., Kambhampati, S., & Scheutz, M. (2011). Planning for agents with changing goals. *International Conference on Automated Planning and Scheduling*, 71-74.

Tate, A. (1990). A review of AI planning techniques. *Readings in planning*, 26-49.

Wikipedia. (2020, Mar. 3). *2018 lower Puna eruption*. (Wikimedia Foundation Inc.) Retrieved Jan. 31, 2020, from https://en.wikipedia.org/wiki/2018_lower_Puna_eruption

Williams, K., & Burdick, J. (2006). Multi-robot boundary coverage with plan revision. *IEEE International Conference on Robotics and Automation (ICRA)*, 1716-1723.

Zacharias, G. L. (2019). *Autonomous Horizons The Way Forward.* Maxwell AFB, Alabama: Air University Press.