The AAAI 2017 Spring Symposium on
Computational Context: Why It's Important, What It Means, and Can It Be Computed?
Technical Report SS-17-03

Machine Learning Approach for Task Generation in Uncertain Environments

Luke Marsh,[1] Iryna Dzieciuch,[2] Douglas S. Lange[2]

[1] Defence Science and Technology Group
Edinburgh, SA, Australia
luke.marsh@dsto.defence.gov.au
[2] Space and Naval Warfare Systems Center Pacific
San Diego, CA, USA
iryna.dzieciuch@navy.mil

Abstract

The command and control of unmanned vehicles is a cognitively intensive task for human operators. Efficient and successful operator performance often depends on a multitude of parameters, such as training, human abilities/factors, timing and situational awareness. Humans are required to multitask in an uncertain environment, process situational data and be able to efficiently utilize autonomous agents in multiple regions of interest. These requirements quite often result in information overload which has consequences on the success of the mission. This is currently an unsolved problem and calls for greater optimization and automation of the command and control of unmanned vehicles.

The cooperative control of unmanned agents in uncertain environments has been a challenge (S. J. Rasmussen, et al). Many models rely on continuous-time, state-space searches in decision trees that are used for planning and execution of the mission. The methods have a number of benefits, however, this approach requires optimization of coordination of play states, by learning from environment's temporal and spatial patterns. In addition, the rate of events in the uncertain environment is always changing, making play models inefficient under a high load of events. This paper attempts to define a space of possible models used in uncertain environments under different levels of complexity, through optimization of assignment coordination.

Introduction

The Intelligent Multi-UxV Planner with Adaptive Collaborative Control Technologies (IMPACT) system is a collection of technologies with the purpose of aiding users in the command and control of multiple Unmanned Vehicles to achieve various tasks (Rowe et al., 2015). The IMPACT system models Unmanned Vehicles, their sensors, and the environment in which they operate in. The unmanned vehicles can be tasked via "plays" to perform certain actions such as scanning and tracking other objects in the simulation. Given the potentially large number of unmanned vehicles in the simulation, human operators can become overwhelmed attempting to maintain situational awareness and control over the vehicles. To address this problem, a Task Manager module has been developed for IMPACT with the aim of assisting human operators (Lange et al., 2014, Gutzwiller et al., 2015). However at times even with the assistance of the task manager, human operators can be overloaded in managing and creating tasks in the system. Additionally, the IMPACT task manager can only assist the operator if it has a representation relating the operational context with the task space being maintained.

The task space is a discrete space that we know how to manage. Complications arise when trying to relate the continuous multivariable space of UxV operations and the continuous space of operator attention with the discrete task space. In this paper, we will outline the task manager module of IMPACT, we describe approaches to discretize the operational context relative to the selection of task types to instantiate for the task manager to manage on behalf of the operator. We will also outline initial efforts to apply machine learning techniques to automatically generate tasks for the task manager, and discuss a model to optimize the scheduling and queuing of tasks under different complexity levels in IMPACT in order to reduce the cognitive load on human operators. The final section concludes the paper and outlines the future effort planned for this work.

Task Manager

Task manager assistance ranges from attention management services to automated control. The core of the task manager is a task model which represents information about the human as well as an intelligent assistant's ability to perform tasks. The tasks may include assigning missions to Unmanned Vehicles, evaluating the performance of a mission against some criteria, making decisions at checkpoints in the mission, or changing the resources available to a team (Lange et al., 2014). If the system detects that the human is overloaded and the computer has the ability to perform a task, the computer can assume responsibility of that task in order to assist the human operator (Lange et al., 2014). Figure 1 represents an example task model which illustrates that there can be multiple methods of achieving a task and one must be selected. Tasks represent elements that must be performed in order to execute the method, and they can be performed either by a human operator or by the computer depending on the actor's capabilities and availability. Methods are alternative approaches to completing a task. While tasks assist human operators in managing their workload in the simulation, they are required to create their own tasks. This can become onerous, especially when the human operator is overloaded.

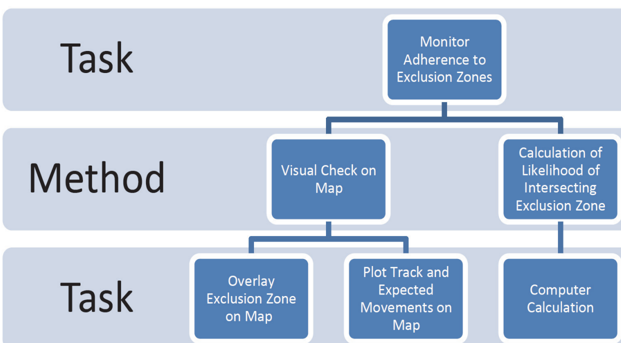

Figure 1. Example Task Model. The recursive task model structure creates alternating levels of tasks and methods.

Machine Learning Task Generation

In this section we will discuss initial ideas and efforts to automatically generate tasks for users of the Task Manager based on machine learning techniques.

All data in the IMPACT simulation is stored in states. The most relevant states in IMPACT include the air, ground, and sea vehicle states which store the live data for all of the vehicles in the simulation. Other data comes from the sensors of the UxVs. This data is stored in camera, video stream, and radio state variables. Each vehicle state is comprised of many variables such as its: current location, velocity, acceleration, current heading, available energy, energy usage rate, list of payloads, and its current tasks. All of this data can be used by machine learning techniques to determine whether a task should be generated.

In using the task manager for IMPACT, a user can create their own task for a job at any point in time. The task could be of any type, and the user can create new types of tasks previously unknown to the system. The trigger for the creation of the tasks can therefore be based on any available information in the simulation. We have no prior knowledge about which data in the simulation was used to make the task. It could be based on any or all information in the system. Therefore the initial machine learning approach taken is the K-Nearest Neighbor algorithm (kNN) (Fix et al., 1951) based on its simplicity and applicability to many problems.

Our high level approach comprises of three steps:

1. Record the states of the IMPACT simulation when a task is created.
2. Continuously monitor the IMPACT simulation states.
3. If the current simulation state matches a state previously recorded when a task was made, then generate this task for the user of the Task Manager.

The core of kNN requires us to determine the distance metrics between the states of the simulation in order to match them and ultimately generate a task. This is not a trivial task as the states of the vehicles in the simulation rarely ever match exactly. The feature standardization technique (Peterson, 2009) is used to remove any bias caused by the state variables having different units and thus different measurement scales.

Initial implementations of the approach have highlighted a number of important design questions: Should the variables in each state in the distance measurement be weighted evenly? Are vehicles considered to be homogenous and thus their states can be compared to another vehicles' state? Should the distance measurements for all individual states be combined or can subsets of states be compared? Finally, what is the threshold for similarity between states for a task to be generated?

Additionally, it is possible to record the simulation state for task creation data for prior simulations and use this data for training and classification. However, what if different numbers and types of vehicles were used in these simulations? Is this data still relevant?

Future work will entail experimenting with different options in attempt to resolve the questions stated above. Given the difficulties in determining the distance metrics between the states for this problem, other more abstract machine learning techniques such as artificial neural networks (ANN) may be more suited to this problem and will be explored. However an ANN is likely to require significant training data of cases in order to train them effectively given the number of variables in each state. Future work will also explore utilizing a time-windowed history of the IMPACT simulation states for task generation.

Task Optimization

In this section we will discuss the complexities of the IMPACT system which can result in the human operator suffering information overload and outline a model to optimize the scheduling and queuing of tasks in IMPACT with the aim of reducing their cognitive load.

Data Collection

All raw data collection from the IMPACT system yields two basic types of data: user generated chat messages and sensor data. Messages from the user come in the form of natural language and can be broken down into four elements: time of message, location, duration and asset. These representations of data are summarized in Table 1.

Time	*Point (ROI)*	*Duration (time)*	*Asset (Sensor, UxV)*
45:00	NAI - 4	4 min	Listening Post / Observation Post (LPOP)
)0:00:00	Ammo Dump	40 min	Imagery
00:17:00	Chow Hall	infinity	360 degrees (imagery)
20:00	Gate 2	10 min	Force (vehicle)
30:00	Gate 2	5 min	Force (vehicle)
message arrival	Point Quebec	15 min	Force (vehicle)
53:00	Barracks	infinity	360 degrees (imagery)
message arrival	Current collation	3 min	Flight line
message arrival	Current location	infinity	UGV, 53 % fuel left

Table 1: Example of the key information obtained from chat messages

The *Time* column in Table 1 represents the time stamp when the message was issued or when the action is planned to take place. The *Point* column represents the region of interest (ROI) where asses is planned to appear. *Duration* is the time of job completion when the trigger is lifted. *Asset* is the name of the sensor or vehicle that is the capability which is used to complete a task.

The idea behind structured chat message retrieval from a database is to quickly distinguish important semantics from the natural language into computer readable form. This stage requires further development and will not be covered in this paper.

Complexity in IMPACT

For the user who simultaneously controls a number of autonomous agents, complexity means higher rates of multitasking. In this case, complexity reduction happens when many simpler tasking states are grouped together in the relevant sequence.

Timing also plays an important role, as events occur at a random rate. For example, the tasking system can be doing a routine task, such as ground monitoring, but if we introduce additional information or events, the user's attention will be diminished. What complexity means for each individual user remains a topic of separate investigation; however, for demonstration of our approach, tasks can be linguistically categorized qualitatively as low, medium and high. Because accumulation of simple tasks at a high rate is overwhelming, timing and rate of task occurrence in the user task queue is a key factor.

Environmental events

Environmental events take place outside of the user's control. These events trigger a user's reaction which will require actions in the IMPACT system to respond to the environmental events. Example environmental events include: Gate runner, Mortar fire, and a user's observation of a chat message. These events have pre-programmed scenario plays and quick reaction responses that are evoked and monitored by user in the IMPACT system.

Sensor Data

Some of the variables in the IMPACT system include data that is supplied by a sensor from the unmanned vehicles (UxV). UxV's operate in a time and space domain and carry variable sensor performance characteristics, for example: Airspeed, Energy Rate, Altitude, and Latitude/Longitude coordinates, etc. The user is constantly updated with sensor information as he or she performs data retrieval when a chat message query is issued. The data representation is summarized in Table 2 below, but it is also composed of the time and ROI information, as well as the duration for the task completion, UxV status, sensor status, and vehicle status.

Time	*Point (lat/long/alt)*	*Duration (time)*	*Sensor Characteristics*
00:00	30.471585; 87.181458; 650	0	Airspeed: 300 Energy rate: 100 Pitch Angle: 0.5 Sensor Type: IR camera
50:00	30.446; 87.150706; 0	50/100 complete	Airspeed: 23 Energy rate: 98 Pitch Angle: 0.5 Sensor Types: IR camera

Table 2: Example of UxV sensor data in the IMPACT system.

Play

Play is a method that the user performs to control unmanned vehicles in the IMPACT system. It usually consists of a pre-programmed number of simpler actions that run their course, and monitored by the user. User tasks are either triggered by a chat message or selected among a list of suggestions.

Common attributes of the data presented in the summary above are space and time. Both the sensors and the IMPACT operators see information in the space and time domain. All events and tasks occur at a specific ROI and a point in time. Suppose, the basic problem we are considering is the state space S of time and location of all variables: sensor, chat (user) and environmental events. The control space is composed of the sequence of decisions in the tasking domain C. The data-task optimization problem of the IMPACT system can thus be states as follows: What is the least complex sequence of tasks that needs to take place to satisfy success of the outcome within a specified completion time?

We can represent state space for the chat message variable as $X = \{x1, x2, x3, x4\}$, where $x1$ is time, $x2$ is point, $x3$ is duration, and $x4$ is asset. Similarly, $Y = \{y1, y2, y3, y4\}$ is the sensor data with similar arguments. Given that currently, one can control a number of UxVs, the representation of these can be written down as $X1$ and $Y1$ for UxV1. The events that take place during a scenario can be represented as $E = \{e1, e2, e3, e4\}$ that describes time, location, duration and type of event. These events trigger a sequence of suggested tasks C that the user can do to reach a favorable outcome under different levels of complexity. For the purpose of illustration for the variable complexity tasking, we propose to use three main variations of complexity settings, as high, medium and low. For example: High complexity is when there are 5 or more events in the tasking queue that require user attention, medium is when there are 3 events that requires user attention, and low is when there are less than 3 events that requires user attention.

For example, sequence of tasking decisions C by operator after a single event E is the following:

1. Send 2 vehicles with cameras to investigate damage - States: X1, Y1, Y2.
2. Send emergency vehicles to point Alpha - States: X1, Y2.
3. Inspect closed roads for blockages or obstacles - States: X1, Y2.
4. Continue until next message - States X2.

In the occurrence of a single event, the set of rules is straightforward but in the occurrence of simultaneous events the situation becomes rather complex. Imagine the user's reaction to multiple emergencies. That is why it is essential to be able to distinguish and prioritize a sequence of decisions or tasks in light of different complexity levels. The control problem can be formulated as follows: Optimize the use of the information available in space and time to find such a sequence of decisions C that gives the maximum result (successfully completed tasks) under varying complexity levels (CL) low medium and high. Thus, the main problem of how to properly control and evaluate states under different complexities becomes a minimization-maximization problem.

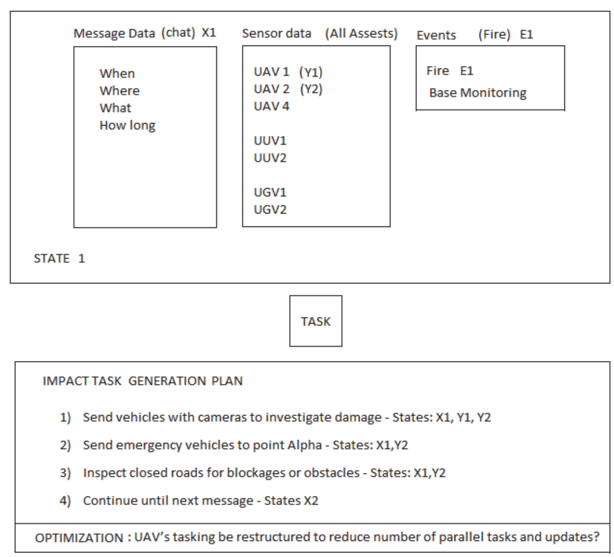

Figure 2: Proposed model for decision making process

The proposed IMPACT planning problem can be generalized to the optimal selection of tasks *max C*, given a message state space X, and sensor data Y and event E, for a complexity level CL. The optimal scheduling and queuing of tasks is meant to reduce (minimize) complexity in the time and space domain. Mathematically it can be described as:

$$\min CL\,(t, s)\ [\max C\,(t, s)\,(X1, \ldots, Xn) \cup (Y1, \ldots, Yn) \cup (E1, .., En)]$$

Thus, decision trees have been realized for the IMPACT system by creating subcategories. The repetitive decision for variable complexity can be described structured in the decision table. For example, are there any chat messages and/or sensor data in the 10 minute window period that overlaps in the time and space domain? How does this information influence the tasking state? Can an UxV's tasking be restructured to reduce the number of parallel tasks and updates? Thus, we need to construct the decision table for each case, an example is shown in Table 3.

Time	*UxV1*		*UxV2*		*Ground Patrol 1*		*Ground Patrol 2*	
	X1	*Y1*	*X2*	*Y2*	*X*	*Y*	*X*	*Y*
00:10	X						X	X
00:14		X	X					
00:21					X	X		

Table 3: Example decision table. X and Y represent chat and sensors which provide data at certain time intervals in the simulation.

After constructing an optimized tasking table we can observe behavior of the separate UxVs and their task load in the time and space domain. Upon closer examination, we observe that two chat messages sent at time 00:10 involve UxV1 and one Ground Patrol 2. At time 00:14 (or 4 seconds later) sensor data from UxV1 and message at UxV2 is sent during Task1. Now we can group chat messages data and sensor data that happens at the same time or are close in space. Next, we repeat the same procedure for consecutive Task 2. Clustering sensor and chat message status is the first step that can be taken towards reducing the operational complexity for the user and to investigate if such data can be used to model the control system under different complexity levels. Such an approach, we believe, will help to optimize tasking decisions and reduce complexity for the human operator. During this step we can use existing data feeds and group tasks in the time and space domain.

Some of the clustering techniques, such as k-nearest neighbor classifier will help us understand dynamics task creation. In the case of the increased complexity described above, when there are four vehicles involved in performing a task for two types of events decreases number of tasks in the queue.

Conclusion

The command and control of unmanned vehicles in the IMPACT system is a cognitively intensive task. Human operators of the system are required to multitask in an uncertain environment, process situational data and be able to efficiently utilize autonomous agents under different levels of complexity and in multiple regions of interest. These requirements can often result in information overload which has consequences on the success of their missions.

In this paper we have outlined initial efforts to apply machine learning techniques to automatically generate tasks for operators of IMPACT and have outlined a model to optimize the scheduling and queuing of tasks in IMPACT with the aim of reducing the cognitive load on human operators. Future work will include restructuring existing data feeds in the time and space domain that can then be effectively used for optimization, and machine learning purposes, continue the research of applying machine learning techniques to automatically generate tasks in the system, and explore the utility of optimizing the scheduling of tasks in order to further reduce the cognitive load on human operators.

References

Fix, E.; Hodges, J.L. 1951. Discriminatory analysis, nonparametric discrimination: Consistency properties. Technical Report 4, USAF School of Aviation Medicine, Randolph Field, Texas.

Gutzwiller, R. S.; Lange, D. S.; Reeder, J.; Morris, R. L.; and Rodas, O.; 2015. Human-computer collaboration in adaptive supervisory control and function allocation of autonomous system teams. In *Proceedings of the 7th International Conference on Virtual, Augmented and Mixed Reality,* 447-456. Los Angeles, Calif.: Springer International Publishing.

Lange, D. S.; Gutzwiller, R. S.; Verbancsics, P.; and Sin, T. 2014. Task models for human-computer collaboration in supervisory control of teams of autonomous systems. In *Proceedings of the 2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support*, 97-102. San Antonio, USA: IEEE.

Lange, D. S.; Verbancsics, P.; Gutzwiller, R.; and Reeder, J. 2012. Command and Control of Teams of Autonomous Units. In Proceedings of the 17th International Command & Control Research & Technology Symposium, Fairfax, VA.: International Command & Control Research & Technology Symposium.

Peterson, L. E. 2009. K-nearest neighbor. *Scholarpedia* 4(2): 1883.

Rowe, A.; Spriggs, S.; and Hooper, D. 2015. Fusion: a framework for human interaction with flexible-adaptive automation across multiple unmanned systems. In Proceedings of the 18th symposium on aviation psychology, 464-469. Dayton, Ohio.: International Symposium on Aviation Psychology.

S. J. Rasmussen, T. Shima, J. W. Mitchell, A. G. Sparks and P. Chandler, State-space search for improved autonomous UAVs assignment algorithm," *2004 43rd IEEE Conference on Decision and Control (CDC).*