Human-UAV Teaming in Dynamic and Uncertain Environments

Demonstration

Alper Turan Alan, Chang Liu, Elliot Salisbury, Stephen D. Prior, Sarvapali D. Ramchurn
University of Southampton
Southampton, UK
[a.t.alan,e.salisbury,chang.liu,sdp,sdr1]@soton.ac.uk

Feng Wu
University of Science and Technology of China
Anhui, China
wufeng02@ustc.edu.cn

Kerry Tatlock and Gareth Rees
MBDA Systems
Bristol, UK
[kerry.tatlock,gareth.rees]@mbda-systems.com

ABSTRACT

In this demonstrator we show how an algorithm developed for human-agent coordination can be used to coordinate human actors on the ground and unmanned aerial vehicles in a rescue mission. A video can be found here: http://goo.gl/QLQD7q.

ACM Reference Format:
Alper Turan Alan, Chang Liu, Elliot Salisbury, Stephen D. Prior, Sarvapali D. Ramchurn, Feng Wu, and Kerry Tatlock and Gareth Rees. 2018. Human-UAV Teaming in Dynamic and Uncertain Environments. In *Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), Stockholm, Sweden, July 10–15, 2018,* IFAAMAS, 3 pages.

1 INTRODUCTION

In this demonstrator, we present the application of a human-agent coordination algorithm presented in [3]. The domain considered is a hostage rescue situation in a dynamic and partially known environment. In such situations, operatives on the ground work in collaboration with tactical commanders at a base to formulate a rescue mission plan. This involves a number of key steps:

(1) Situational awareness: typically data from previous missions, satellite imagery, and live camera feeds from unmanned aerial vehicles can be used to gather such information. This process results in a map of where the threats might be, which might be the safest routes, where the hostage might be located, and whether the mission is feasible in the time allocated. This step also helps specify what information is missing and what course of action might be taken to gather this missing information.
(2) Mission Planning: once all the information about the environment has been gathered, the planning process involves creating a path for each of the human and autonomous assets (on the ground or in the air) for them to gather information and rescue the hostage.
(3) Mission Execution: Given the initial plan, both human and autonomous actors undertake their tasks and continuously share what they see on the ground (e.g., camera feeds from UAVs or audio from human actors) in order to maintain situational awareness at all times. The mission is coordinated by tactical commanders at a base who both control the UAVs and orietentate the operatives on the ground.

Proc. of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2018), M. Dastani, G. Sukthankar, E. André, S. Koenig (eds.), July 10–15, 2018, Stockholm, Sweden.

This high tempo situation typically gets very complex very quickly as new threats are detected, equipment is damaged by threats, or operatives are injured or tire out. Crucially, the confusion created by such a dynamic situation can lead to poor judgement and the failure of the mission. Given this, it is not only important to develop algorithms to support tactical decision making but also interaction mechanisms that ensure that the commanders and operatives are not burdened by the need to control unmanned vehicles at all times, and hence devote their attention to mission planning, quick reaction to threats, and the most effective execution of the mission.

Against this background, in this project (funded by DSTL), we developed a multi-UAV coordination system building upon [2] and successfully tested the system out in the field in a simulated hostage situation. The system involves the implementation of:

(1) Interfaces for tactical planners to manage a mission and an interface for operatives on the ground to receive instructions in real-time and communicate back to base.
(2) A scalable back-end framework that is able to cope with the addition of both human and UAV assets and manage the communication of messages among them. The platform is able to accommodate heterogeneous UAVs and human actors with different capabilities and communication devices.
(3) A human-UAV coordination algorithm based on [3] that can be used (via the interfaces specified) to generate coordinated paths for both human and unmanned assets.
(4) A Slam-based UAV navigation system for GPS denied environments.

The system is evaluated in the real world in a scenario involving two DJI phantom III UAVs and a custom-built UAV for GPS-denied environments. The mission was executed within dedicated University facilities (with flight permissions) and the tactical command was set up using a specially engineered vehicle with communication capabilities. The system was evaluated by experts from a defence company and successfully achieved the key objectives of the project.

2 HUMAN-UAV COORDINATION MODEL

Here we describe the problem of coordinating humans and UAVs in a dynamic and uncertain environment using a Partially Observable Markov Decision Process (POMDP). We assume that the soldiers and UAVs know their current locations using GPS. Hence the states of the operatives and UAVs are fully observable to a planner agent. However, the state of the environment is hidden to the agent and therefore UAVs are deployed to collect information about it. In our setting, each UAV has sensors that can record the reading of the environmental factors at its current location and hence the state of the environment is partially observable to the planner agent.

At each time step, the planner agent first selects an action for the soldier and a joint action for the UAVs. As received from the agent, the actions are executed by the soldier and the UAVs. This results a state transition in the world and an observation about the state of the environment is received by the UAVs. After that, the UAVs send back their current state and observation to the agent and the soldier also reports her current state. Then, the agent proceeds to the next time step and the process repeats until the planning horizon is reached. Thus, the goal is to find actions that ensure that UAVs gather as much information as possible to maximize the number of tasks that the soldier is able to complete without being harmed by the environment.

To address those challenges, we developed the OPAS algorithm in [3] — Online Planning with Active Sensing. In each time step of the algorithm, the soldier computes a policy and specifics her action and the UAVs use this policy to update their sensing action. In turn, for the soldier, she can update and enhance the policy she computes based on information reported by UAVs as they fly over the areas chosen. In more details, we use simulation-based task planning to compute the policy for the soldier and choose the sensing action for the UAVs based on the Value of Information (VoI) — a well-known concept for information theory.

3 SYSTEM ARCHITECTURE

Now, to implement the above algorithm within a real-world environment requires the development of hardware and software tools for humans to interact with the UAVs as well as to generate plans for both soldiers and UAVs. To this end, we developed a number of elements that ensure that the system can scale to large numbers of both humans and machine actors. On the hardware side, communications was provided by a wifi router and computation was done on a web server and laptops and mobile phones were used to run the interfaces. The main software components include:

1. Message Queue (MQ): this is initialised on a server to pass messages between interfaces and the applications used to communicate with soldiers or with UAVs.

2. Tactical command Interface: this (shown in the video) contains a number of UI elements (a map and controls) that allow the commanders to formulate a plan and send it to the soldier and UAVs.

3. Soldier App : this resides on the phone used by the soldier. It uses the GPS location of the phone and communicates it to a server (after first registering with it). The app then listens to messages from MQ that are directed at it and displays it on the phone. If the soldier wishes to send a message to the commanders, a chat window can be opened to do so.

4. UAV App: this is an app developed using the DJI Android SDK[1]. Thus we augmented the basic app provided by the SDK to listen to messages from MQ and act on them. This provided an easy way of remotely controlling any DJI Phantom UAV that is controllable by this Android App.

Having developed these tools, the planning agent (i.e., using the OPAS algorithm) was implemented on the server and provided with real-time positioning information as well as a grid-based definition of the state of the world. Hence, for the demonstrator, we fixed the domain where the algorithm was to be deployed and simulated obstacles (i.e., buildings) by marking parts of the grid as unpassable by the soldiers. Moreover, we simulated threats (that could harm soldiers and UAVs) that are detected by the UAVs. These were provided as sensed inputs (as if coming from the UAVs) to the planner agent in order to regenerate a workable plan.

[1]https://developer.dji.com/api-reference/android-api.

4 HUMAN-UAV INTERACTION

A key part of the demo involved showing how the both software agents and robotic agents can be made to work in partnership with the soldiers and tactical commanders. Hence, the interaction between them was engineered to maximise understanding between human and machine-based actors.

Plans computer by the planner agent were displayed on a screen in terms of suggested paths for each actor, along with the ability to modify such paths. This builds upon prior the interface designed in [1, 2]. Furthermore, once approved by the tactical commander, the paths were passed on to the soldier app through which the soldier confirmed that she was going to enact the plan.

As expected, using only the apps to coordinate soldiers did not work very well and detracts from the main goal of the system to reduce the need for humans to waste time communicating instructions or checking the plan. Hence, we provided both soldiers on the ground and tactical commanders with walkie talkies. This allowed them to quickly confirm plans they could see on the apps and react quickly to changes that came through.

Finally, we also provided the ability to the tactical commanders to take over the the control of individual UAVs by providing controls on the apps to do so. This was particularly important to manage the indoor UAV we describe next.

5 INDOOR NAVIGATION UAV DESIGN

The custom built, 6.5 inch (propeller size) small quadrotor UAV was developed for autonomous operation without complete dependency on the GNSS services. It allows the UAV to maintain full capabilities in GPS/GNSS - denied environments, such as indoor spaces or situations when GNSS service failed.

The UAV is equipped with Pixracer autopilot running px4 flight stack. In addition to the native inertial sensors on the Pixracer autopilot, it receives redundant localisation measurements from three positional sensors to realise accurate and robust manoeuvring.

The additional onboard sensors include: one GNSS global positioning system measuring the geo-referenced location; one PX4flow downward facing smart camera measuring the horizontal velocity with respect to the vehicle body, and one forward-facing Intel Euclid Simultaneous Localisation And Mapping (SLAM) local positioning sensor measuring relative position and orientation of the vehicle. Additionally, the Intel Euclid provides depth sensing for obstacle detection and depth map building, which provides the information for potential ability for obstacle avoidance.

The Pixracer receives command and update flight status through onboard radio telemetry, while a ground station computer coordinates between the command/status to/from the radio telemetry and main server computer. The ground station computer communicate with the main server computer through a WIFI router. In addition, this small quadrotor also carries an onboard thermal camera for heat maps or human detection.

REFERENCES

[1] Sarvapali Ramchurn, Joel Fischer, Yuki Ikuno, Feng Wu, Jack Flann, and Antony Waldock. 2015. A Study of Human-Agent Collaboration for Multi-UAV Task Allocation in Dynamic Environments. In *International Joint Conference on Artificial Intelligence* (2015-01-01). http://eprints.soton.ac.uk/377185/

[2] Sarvapali Ramchurn, Edwin Simpson, Joel Fischer, Trung Dong Huynh, Yuki Ikuno, Steven Reece, Wenchao Jiang, Feng Wu, Jack Flann, S.J. Roberts, Luc Moreau, T. Rodden, and N.R. Jennings. 2016. A Disaster Response System based on Human-Agent Collectives. *Journal of Artificial Intelligence Research* 57 (2016), 661–708. http://www.jair.org/media/5098/live-5098-9699-jair.pdf

[3] Feng Wu, Sarvapali D. Ramchurn, and Xiaoping Chen. 2016. Coordinating human-UAV teams in disaster response. In *International Joint Conference on Artificial Intelligence (IJCAI-16)* (2016-04-01). 524–530. http://eprints.soton.ac.uk/393725/