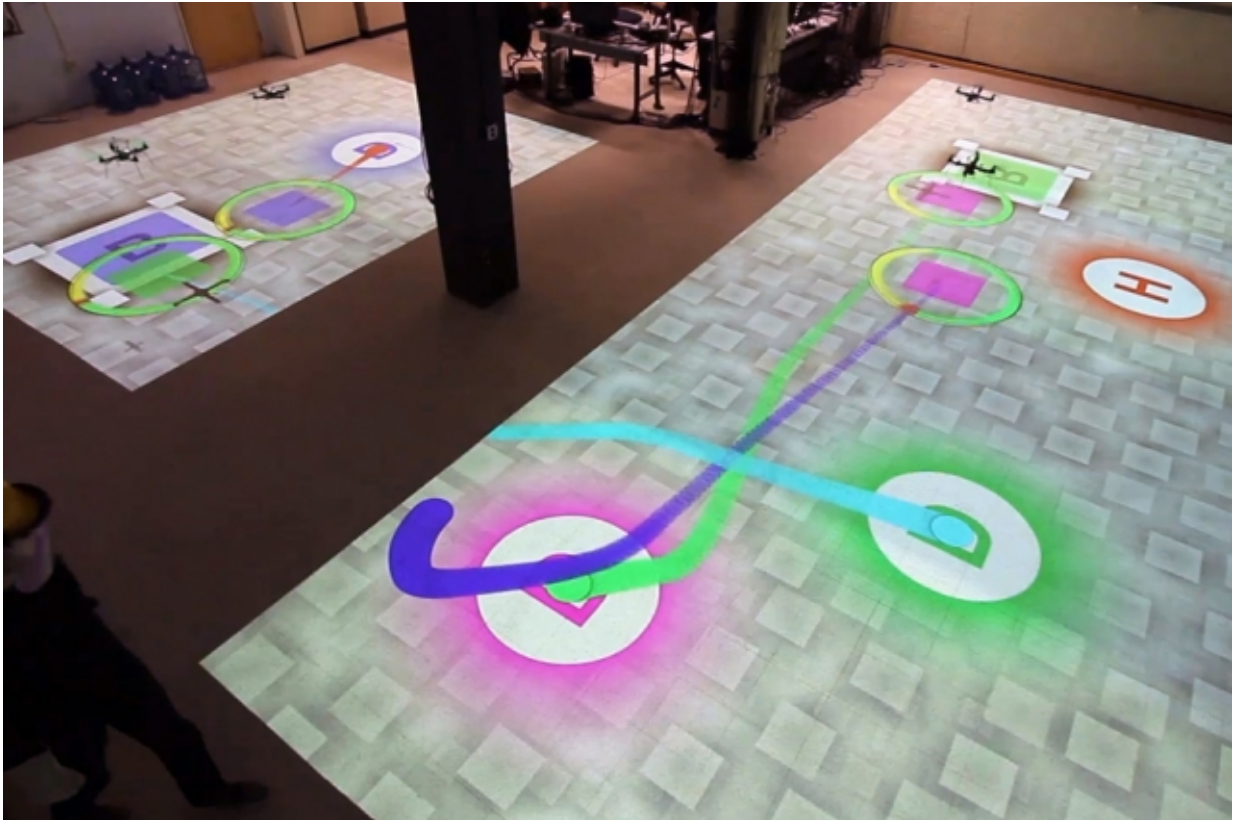# Helping robots handle uncertainty

June 3 2015, by Larry Hardesty

Researchers are using a small group of robotic helicopters to address the problem of drone package delivery. The scenario includes base stations (marked with a "B") and destinations (marked with a "D"). The colored lines represent the planned paths of the 4 helicopters. Credit: Shayegan Omidshafiei and Shih-Yuan Liu

Decentralized partially observable Markov decision processes are a way

to model autonomous robots' behavior in circumstances where neither their communication with each other nor their judgments about the outside world are perfect.

The problem with Dec-POMDPs (as they're abbreviated) is that they're as complicated as their name. They provide the most rigorous mathematical models of multiagent systems—not just robots, but any autonomous networked devices —under uncertainty. But for all but the simplest cases, they've been prohibitively time-consuming to solve.

Last summer, MIT researchers presented a paper that made Dec-POMDPs much more practical for real-world robotic systems. They showed that Dec-POMDPs could determine the optimal way to stitch together existing, lower-level robotic control systems to accomplish collective tasks. By sparing Dec-POMDPs the nitty-gritty details, the approach made them computationally tractable.

At this year's International Conference on Robotics and Automation, another team of MIT researchers takes this approach a step further. Their new system can actually generate the lower-level control systems from scratch, while still solving Dec-POMDP models in a reasonable amount of time.

The researchers have also tested their system on a small group of robotic helicopters, in a scenario mimicking the type of drone package delivery envisioned by Amazon and Google, but with the added constraint that the robots can't communicate with each other.

"There's an offline planning phase where the agents can figure out a policy together that says, 'If I take this set of actions, given that I've made these observations during online execution, and you take these other sets of actions, given that you've made these observations, then we can all agree that the whole set of actions that we take is pretty close to

optimal,'" says Shayegan Omidshafiei, an MIT graduate student in aeronautics and astronautics and first author on the new paper. "There's no point during the online phase where the agents stop and say, 'This is my belief. This is your belief. Let's come up with a consensus on the best overall belief and replan.' Each one just does its own thing."
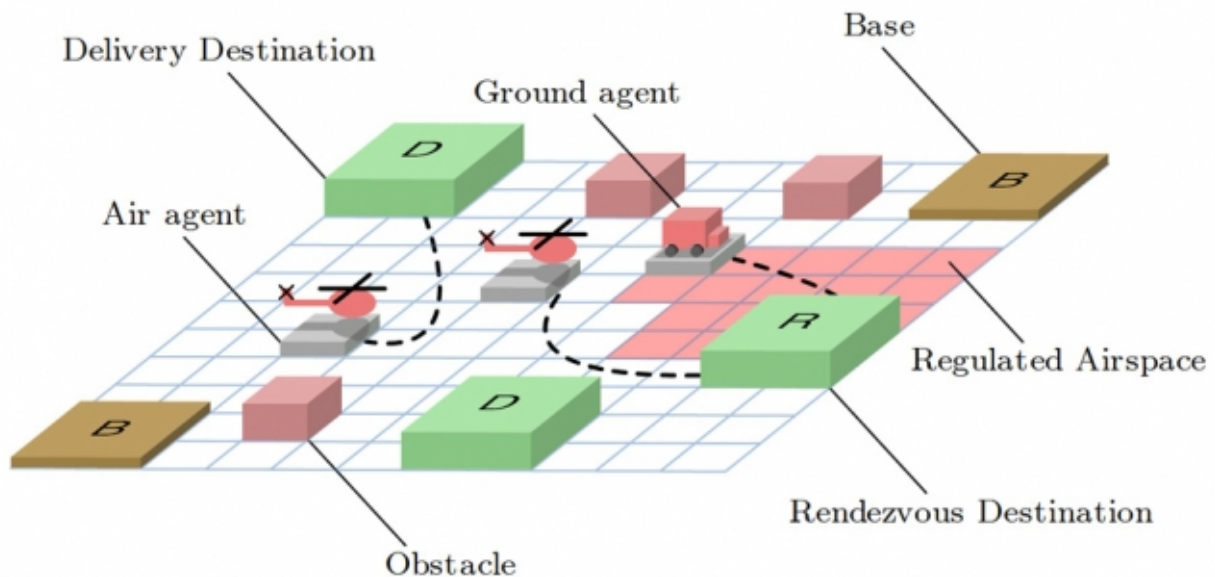
This diagram shows the various destinations and obstacles in the researchers' scenario. Credit: Shayegan Omidshafiei

What makes Dec-POMDPs so complicated is that they explicitly factor in uncertainty. An autonomous robot out in the world may depend on its sensor readings to determine its location. But its sensors will probably be slightly error-prone, so any given reading should be interpreted as

defining a probability distribution surrounding the actual measurement.

Even an accurate measurement, however, may be open to interpretation, so the robot would need to build a probability distribution of possible locations on top of the probability distribution of sensor readings. Then it has to choose a course of action, but its possible actions will have their own probabilities of success. And if the robot is participating in a collaborative task, it also has to factor in the probable locations of other robots and their consequent probabilities of taking particular actions.

Since a probability distribution consists of a range of possible values—in principle, an infinite number of values—solving a problem with probabilities heaped on probabilities is much harder than solving a problem involving discrete values.

To make it easier to solve a Dec-POMDP, Omidshafiei and his co-authors—his thesis advisor, Maclaurin Professor of Aeronautics and Astronautics Jonathan How; Ali-akbar Agha-mohammadi, a former postdoc in MIT's Laboratory for Information and Decision Systems who is now at Qualcomm Research; and Christopher Amato, who led the earlier work on Dec-POMDPs as a postdoc in MIT's Computer Science and Artificial Intelligence Laboratory and has just joined the faculty of the University of New Hampshire—decompose it into two problems, both of which involve graphs.

A graph is data representation consisting of nodes, usually depicted as circles, and edges, usually depicted as lines connecting the circles. Network diagrams and family trees are familiar examples.

The researchers' algorithm first constructs a graph in which each node represents a "belief state," meaning a probabilistic estimate of an agent's own state and the state of the world. The algorithm then creates a set of control procedures—the edges of the graph—that can move the agent

between belief states.

The researchers refer to these control procedures as "macro-actions." Because a single macro-action can accommodate a range of belief states at both its origin and its destination, the planning algorithm has removed some of the problem's complexity before passing it on to the next stage.

For each agent, the algorithm then constructs a second graph, in which the nodes represent macro-actions defined in the previous step, and the edges represent transitions between macro-actions, in light of observations. In the experiments reported in the new paper, the researchers then ran a host of simulations of the task the agents were intended to perform, with agents assuming different, random states at the beginning of each run. On the basis of how well the agents executed their tasks each time through, the planning algorithm assigned different weights to the macro-actions at the nodes of the graph and to the transitions between nodes.

The result was a graph capturing the probability that an agent should perform a particular macro-action given both its past actions and its observations of the world around it. Although those probabilities were based on simulations, in principle, autonomous agents could build the same type of graph through physical exploration of their environments.

Finally, the algorithm selects the macro-actions and transitions with the highest weights. That yields a deterministic plan that the individual agents can follow: After performing macro-action A, if you make measurement B, execute macro-action C.

  More information: "Decentralized Control of Partially Observable Markov Decision Processes using Belief Space Macro-actions." www.cs.unh.edu/~camato/publica … Action-ICRA-OAAH.pdf

*This story is republished courtesy of MIT News
(web.mit.edu/newsoffice/), a popular site that covers news about MIT
research, innovation and teaching.*

Provided by Massachusetts Institute of Technology