DOI:10.1145/1538788.1538811

Technical Perspective
The Ultimate Pilot Program

By Stuart Russell and Lawrence Saul

IN ONE SCENE from "The Matrix," two leaders of the human resistance are trapped on the roof of a skyscraper. The only means of escape is by helicopter, which neither can operate. The humans quickly call up a "pilot program" for helicopter flight, absorb the knowledge instantly via a brain-computer interface, and take off in the nick of time.

The following paper by Coates, Abbeel, and Ng describes an equally remarkable feat: learning to fly helicopter aerobatics of superhuman quality by watching a few minutes of a human expert performance. Before you read the paper, we suggest watching the videos at http://heli.stanford.edu/.

The authors provide careful descriptions of the problem and of the technical innovations required for its solution. The paper's importance lies not only in these innovations, but also in the way it illustrates the flavor of modern artificial intelligence research. AI has grown to encompass, in a seamless way, techniques from areas such as statistical learning, dynamical systems, and control theory, and has reintegrated with areas such as robotics, vision, and natural language understanding that many thought had gone their own way. The key to reunification has been the emergence of effective techniques for probabilistic reasoning and machine learning. The authors illustrate this trend perfectly, solving a problem in robotics that had resisted traditional control theory techniques for many years.

Learning to fly a helicopter means learning a policy—a mapping from states to control actions. What form should the mapping take and what information should be supplied to the learning system? Some early work adopted the idea of observing expert performance to learn to fly a small plane,[2] using *supervised* learning methods and representing policies as decision trees. In this approach, each expert action is a positive example of the function to be learned, while each action not taken is a negative example. Unfortunately, the resulting policies fail miserably when any perturbation puts the aircraft into a state not seen during training. Perhaps this is not surprising, because the policy has no idea how the vehicle works or what the pilot is attempting.

In contrast, the authors formulate the problem as a Markov decision process (MDP), where the *transition model* specifies how the vehicle works, the *reward function* specifies what the pilot is trying to do, and the *optimal policy* maximizes the expected sum of rewards over the entire trajectory. Initially, of course, the transition model and reward function are unknown, so the learning system cannot solve for the optimal policy. In the well-established setting of *reinforcement learning*, the learning system acts in the world and observes outcomes and rewards. For many problems, learning a model and a reward function requires fewer experiences than trying to learn a policy directly—and experiences are always in short supply in robot learning.

Pure *tabula rasa* reinforcement learning is not applicable to helicopter aerobatics, however, for two reasons: First, in the early stages of learning there would be far too many crashes; second, the reward function is not known even to the experimenters, so a reward signal cannot easily be provided to the learning system. The *apprenticeship learning* setting adopted by the authors avoids both problems by learning from expert behaviors.

By observing the helicopter's trajectory while the expert is flying, the learning system can acquire a transition model that is reasonably accurate in the regions of state space that are likely to be visited during these maneuvers. The role of *prior knowledge* is crucial here; while the model parameters are learned, the model structure is determined in advance from general knowledge of helicopter dynamics.

The task of learning the reward function from expert behavior is called "inverse reinforcement learning." Introduced in AI in the late 1990s, this actually has a long history in economics.[2] For helicopter aerobatics, the reward function specifies what the desirable trajectories are, such that following them yields high reward, and how deviations should be penalized. This information is implicit in the expert's behavior and its variability. To account for this variability, the authors develop a probabilistic generative model for trajectories, borrowing methods from speech recognition and sequence alignment to handle variations in timing. After learning from several expert performances, the reward function actually defines a much better trajectory than the expert could demonstrate, and the autonomous helicopter eventually outperforms its human teacher.

The authors' success in this difficult task reflects fundamental progress in our field. While achieving comparable success on other difficult robotic tasks is not yet a routine application of off-the-shelf methods, the technology of apprenticeship learning provides a plausible template for progress.

References

1. Sammut, C., Hurst, S., Kedzier, D. and Michie, D. Learning to fly. In *Proceedings of the Intern. Conf. on Machine Learning* (1992).
2. Sargent, T.J. Estimation of dynamic labor demand schedules under rational expectations. *J. Political Economy 86* (1978), 1009–1044.

Stuart Russell is a professor of CS, chair of the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley, and co-chair of *Communications*' Research Highlights Board.

Lawrence Saul is an associate professor in the Department of Computer Science and Engineering at the University of California, San Diego, and a member of *Communications*' Research Highlights Board.