From: AAAI-93 Proceedings. Copyright © 1993, AAAI (www.aaai.org). All rights reserved.

AIR-SOAR: Intelligent Multi-Level Control

Douglas J. Pearson, Randolph M. Jones, and John E. Laird
Artificial Intelligence Laboratory
The University of Michigan
1101 Beal Ave.
Ann Arbor, Michigan 48109–2122
dpearson@caen.engin.umich.edu

Autonomous systems must be able to deal with dynamic, unpredictable environments in real time. Our video describes a system for intelligent control of an airplane, within a realistic flight simulator (the Silicon Graphics flight simulator). The simulator allows asynchronous control of the plane's throttle, ailerons, elevator and other control surfaces by an external system, and it provides limited asynchronous sensing of the plane's motion. The result is a highly dynamic, real time domain in which models of the plane (and, potentially, other aircraft) are updated 20 times a second. Control of flight is complex. Unexpected events such as wind or turbulence must be responded to in a timely fashion. Further, identical control movements have different effects depending on the plane's position and environmental conditions, making precise prediction of action effects difficult. The agent must also deal with delays in feedback from its actions, waiting for the plane to respond to changes in the control surfaces. The domain requires simultaneous execution of a range of tasks at different levels of complexity and granularity, from high level maneuvers like takeoff, landing and banked turns to low level tasks such as maintaining altitude, keeping the wings level and controlling the stick.

Our autonomous agent for the flight domain is *Air-Soar* [Pearson *et al.*, 1993]. The agent is built within Soar [Laird *et al.*, 1987], a general problem solving and learning architecture. Soar solves problems by successively applying operators within problem spaces. Air-Soar reasons about flight with five problem spaces, each reasoning at a different level of granularity. In addition, the system achieves and maintains multiple goals simultaneously, both within and across levels. For example, at the highest level the system may be both climbing and turning to a new heading. Across levels, lower-level constraints may be achieved while performing higher-level goals, such as leveling the wings during a climb to a new altitude. Thus, Air-Soar supports *achievement goals*, where the goal is to reach a particular state (such as a new altitude), and *homeostatic goals*, in which constraints must be continuously maintained [Covrigaru and Lindsay, 1991; Kaelbling, 1986]. Homeostatic goals often interact with achievement goals in the flight domain. Examples include keeping the wings level while taking off, and maintaining the current altitude during a turn. Air-Soar must combine the requirements of the different types of goals to make steady progress along a flight path, without losing control by focusing only on a single aspect of the current task (such as only monitoring the altitude during a climb).

Typically, all of Air-Soar's levels are active simultaneously, trying to maintain or achieve their current goals. This hierarchical approach supports reactive behavior at multiple levels of granularity. Rather than explicitly monitoring the fact that all of the plane's flight parameters are within expected ranges, each problem space notices when the values deviate from constraints it is trying to achieve or maintain, and moves to correct them. These corrections produce changes at lower granularity levels, ultimately resulting in stick commands to control the plane.

Sensitivity at different grain sizes means that Air-Soar is able to respond to a wide range of unexpected events. For instance, after completing a turn, the plane might not be perfectly level, causing the heading to slowly change. Although the rate of change is low, after a while Air-Soar would notice the heading was no longer within range and would turn to correct it. Alternatively, if a sudden burst of turbulence caused the plane's wings to vere suddenly, then the system would react to the sudden increase in *turn rate* directly, before the heading had changed enough to be noticed.

Air-Soar is currently able to take off, level off and then follow a pre-set flight pattern including a series of turns and altitude changes, returning to land on (or near) the runway. We have simulated "turbulence" during Air-Soar runs by manually moving the mouse controlling the plane's stick while Air-Soar controls the plane. Air-Soar responds immediately to the "turbulence" and continually attempts to keep the plane on course. If the plane is flying level and is pulled out of level flight by the mouse, Air-Soar recovers by responding to the change in the plane's roll to reestablish level flight. Air-Soar's successful execution of the flight plan, together with our experiments with "turbulence", demonstrates the system's ability to perform robustly in a highly reactive, real-time domain. In addition, it highlights Air-Soar's ability to reason about multiple simultaneous goals at various levels of granularity.

References

[Covrigaru and Lindsay, 1991] Arie A. Covrigaru and Robert K. Lindsay. Deterministic autonomous systems. *AI Magazine*, 12(3):110–117, 1991.

[Kaelbling, 1986] Leslie Pack Kaelbling. An architecture for intelligent reactive systems. In Michael P. Georgeff and Amy L. Lansky, editors, *Reasoning about actions and plans: Proceedings of the 1986 Workshop*, pages 395–410. Morgan Kaufmann, 1986.

[Laird *et al.*, 1987] John E. Laird, Allen Newell, and Paul S. Rosenbloom. Soar: An architecture for general intelligence. *Artificial Intelligence*, 33(1):1–64, 1987.

[Pearson *et al.*, 1993] Douglas J. Pearson, Scott B. Huffman, Mark B. Willis, John E. Laird, and Randolph M. Jones. Intelligent multi-level control in a highly reactive domain. In Charles E. Thorpe F.C.A. Groen, S. Hirose, editor, *Proceedings of the Third International Conference on Intelligent Autonomous Systems*, pages 449–458. IOS Press, 1993.