a constant angular velocity of the image of the environment) over its compound eye (see figure). Consistent with this strategy, a bee utilizes the following simple control laws when approaching a landing site on a flat surface:

1. The optical flow of the surface is held constant throughout the descent.
2. Forward speed is held proportional to vertical speed throughout the descent.

This simple combination of control laws enables a smooth landing with minimal computation. The forward speed and rate of descent are reduced together, and are both close to zero at touchdown. No knowledge or measurement of instantaneous speed or height above the ground are necessary. This combination of control laws can readily be modified for a biomorphic flyer, which has a nonzero stalling speed.

This work was done by Sarita Thakoor of Caltech for **NASA's Jet Propulsion Laboratory** *and by G. Stange, M. Srinivasan, and Javaan Chahl of Australian National University and Butler Hine and Steven Zornetzer of Ames Research Center for the NASA Intelligent Systems Program. Further information is contained in a TSP (see page 1).*
NPO-30545

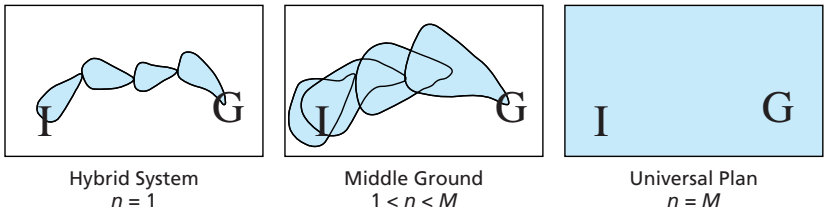
Domain Compilation for Embedded Real-Time Planning

Robustness is increased at the price of a moderate increase in complexity.

NASA's Jet Propulsion Laboratory, Pasadena, California

A recently conceived approach to automated real-time control of the actions of a robotic system enables an embedded real-time planning algorithm to develop plans that are more robust than they would otherwise be, without imposing an excessive computational burden. This approach occupies a middle ground between two prior approaches known in the art as the universal-plan and hybrid approaches.

Ever since discovering the performance limitations of taking a sense-plan-act approach to controlling robots, the robotics community has endeavored to follow a behavior-based approach in which a behavior includes a rapid feedback loop between state estimation and motor control. Heretofore, system architectures following this approach have been based, variously, on algorithms that implement universal plans or algorithms that function as hybrids of planners and executives. In a typical universal-plan case, a set of behaviors is merged into the plan, but the system must be restricted to relatively small problem domains to avoid having to reason about too many states and represent them in the plan. In the hybrid approach, one implements actions as small sets of behaviors, each applicable to a limited set of circumstances. Each action is intended to bring the system to a subgoal state. A planning algorithm is used to string these actions together into a sequence to traverse the state space from an initial or current state to a goal state. The hybrid approach works well in a static environment, but it is inherently brittle in a dynamic environment because a failure can occur when the environment strays beyond the region of applicability of the current activity.



Note: Each rectangle symbolizes state space, and colored areas represent portions of the state space covered by a universal(*n*) plan for target goals from the initial state I to the goal state G.

Coverage of State Space by an *n*-level plan increases with *n*.

In the present approach, a system can vary from the hybrid approach to the universal-plan approach, depending on a single integer parameter, denoted n, which can range from 1 to a maximum domain-dependent value of M. As illustrated in the figure, $n = 1$ represents the hybrid approach, in which each linked action covers a small part of the state space of the system. As n increases, the portion of state space associated with each action and its subgoal grows. When n reaches M, coverage extends over the full state space, so that the system contains a universal plan.

Through incorporation of an embedded real-time planning algorithm that follows this middle-ground approach, a hybrid system can be made much more robust in a dynamic environment. In such a system, the planning algorithm passes the current subgoals (instead of activities) to an executive algorithm. The executive algorithm then uses the real-time planning algorithm to determine when to perform which action until it determines either that the current subgoals have been reached or that they cannot be reached within n steps. If the current subgoals have been reached, the planning algorithm gives new subgoals to the executive algorithm. If it has been determined that the current subgoals cannot be reached, the planning algorithm must alter the sequence of actions.

A structure for finding the next step on an n or fewer step path to a subgoal is called a universal(n) plan. Because complexity increases sharply with n, it is necessary to choose n small enough to avoid an excessive computational burden but large enough that it is possible to make a universal(n) plan that makes the system robust in the sense that it can reach each given subgoal state from any state in a large region of the state space.

This approach involves utilizing knowledge compilation research by implementing an off-line compiler that generates a universal(n) plan from a system description. In the first step of a two-step process, the system description is converted into a logical expression, in what is known as a conjunctive nor-

mal form (CNF) that is based on the effects and preconditions of actions in an *n*-step plan. In the second step, the aforementioned research results are used to convert the CNF representation into a decomposable negation normal form (DNNF) representation. It turns out that the computation time needed to evaluate a DNNF expression to compute an optimal *n*-step plan increases only linearly with the DNNF representation size.

This work was done by Anthony Barrett of Caltech for **NASA's Jet Propulsion Laboratory**. *Further information is contained in a TSP (see page 1).*

The software used in this innovation is available for commercial licensing. Please contact Don Hart of the California Institute of Technology at (818) 393-3425. Refer to NPO-40296.

Semantic Metrics for Analysis of Software

These metrics represent a more human-oriented view of software.

Goddard Space Flight Center, Greenbelt, Maryland

A recently conceived suite of object-oriented software metrics focus is on semantic aspects of software, in contradistinction to traditional software metrics, which focus on syntactic aspects of software. Semantic metrics represent a more human-oriented view of software than do syntactic metrics. The semantic metrics of a given computer program are calculated by use of the output of a knowledge-based analysis of the program, and are substantially more representative of software quality and more readily comprehensible from a human perspective than are the syntactic metrics.

Semantic metrics have the potential to help software engineers identify fragile, low-quality sections of code much earlier in the development process than is possible by use of syntactic metrics. By enabling earlier and better detection of faults, semantic metrics are expected to make maintenance of software less time-consuming and expensive and to make software more reusable. Because it is less costly to correct faults found earlier than to correct faults found later in the software-development process, it is expected that the overall cost of developing software will be reduced. Moreover, because semantic metrics provide better measures of internal documentation descriptiveness (descriptiveness of the comments and identifiers in software), all aspects of development of software can be expected to benefit from improved understanding of the software.

Prototype software called "SemMet" for computing semantic metrics is undergoing development. In SemMet, semantic metrics are described within the context of knowledge-based systems that consist of semantic networks formed from conceptual graphs. Conceptual graphs are often used for semantic networks for natural language processing; however, the use of conceptual graphs is also a general and fairly common knowledge-representation technique. In the computation of semantic metrics, concepts and conceptual relations from conceptual graphs inside a knowledge base are used as input. The output semantic metrics are presented in a report.

This work was done by Letha H. Etzkorn, Glenn W. Cox, Phil Farrington, Dawn R. Utley, Sampson Ghalston, and Cara Stein of the University of Alabama at Huntsville for **Goddard Space Flight Center**.*Further information is contained in a TSP (see page 1).*

In accordance with Public Law 96-517, the contractor has elected to retain title to this invention. Inquiries concerning rights for its commercial use should be addressed to:

Letha Hughes Etzcorn, Ph.D., P.E.
Assistant Professor
Computer Science Department
University of Alabama in Huntsville
Huntsville, AL 35899

Refer to GSC-14752-1, volume and number of this NASA Tech Briefs *issue, and the page number.*

Simulation of Laser Cooling and Trapping in Engineering Applications

This design instrument shows good agreement with experimental measurements.

NASA's Jet Propulsion Laboratory, Pasadena, California

An advanced computer code is undergoing development for numerically simulating laser cooling and trapping of large numbers of atoms. The code is expected to be useful in practical engineering applications and to contribute to understanding of the roles that light, atomic collisions, background pressure, and numbers of particles play in experiments using laser-cooled and -trapped atoms. The code is based on semiclassical theories of the forces exerted on atoms by magnetic and optical fields. Whereas computer codes developed previously for the same purpose account for only a few physical mechanisms, this code incorporates many more physical mechanisms (including atomic collisions, sub-Doppler cooling mechanisms, Stark and Zeeman energy shifts, gravitation, and evanescent-wave phenomena) that affect laser-matter interactions and the cooling of atoms to submillikelvin temperatures. Moreover, whereas the prior codes can simulate the interactions of at most a few atoms with a resonant light field, the number of atoms that can be included in a simulation by the present code is limited only by computer memory. Hence, the present code represents more nearly completely the complex physics involved when using laser-cooled and -trapped atoms in engineering applications.

Another advantage that the code incorporates is the possibility to analyze the interaction between cold atoms of different atomic number. Some properties that cold atoms of different atomic