ODYSSEUS: An Autonomous Mobile Robot (extended abstract)

R. Simmons, S. Thrun, C. Athanassiou, J. Cheng
L. Chrisman, R. Goodwin, G.-T. Hsu, H. Wan

School of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

INTRODUCTION

Odysseus[1] is a small wheeled robot equipped with an arm, sonar sensors and a camera system. It is connected by radio links to a pool of computers that control the robot. A main emphasise in the software implementation of Odysseus' control is the ability to act and react in real-time. Odysseus combines low-level reactive mechanisms with global planning and exception handling, using a wide variety of control and AI techniques, ranging from A* planning and hidden Markov model-based speech recognition to artificial neural networks and reinforcement learning. On the lowest level of behavior the robot employs several fast on-board obstacle detection and avoidance mechanisms for safely operating in unpredictable, dynamic environments. Odysseus' global navigation is map-based. The sonar sensor is used for incrementally constructing a model of its environment. The camera is used for detecting target objects. Odysseus is able to identify and to navigate to particular objects, as well as to explore its environment autonomously in order to gain knowledge. The robot is operated using a speaker-independent speech recognition/generation system. In addition, a graphical interface is used for monitoring the operation of the robot.

HARDWARE

The robot is based on a HERO-2000, a wheeled robot with manipulator and a gripper. There are two main sensor systems mounted on top of the robot: a black and white camera, and sonar sensors. There is one fixed base-mounted sonar and one sonar sensor on the top of the robot that can be directed by a rotating mirror to give a full 360^o sweep. Odysseus operates tetherless: it is powered by rechargeable batteries, and connected to its frontend computers by two radio links, one for the camera, and one for a sequential RS-232 port.

Odysseus is controlled by a distributed system, consisting of 4 SUN SPARC stations, one NeXT station, and a local 8088 microprocessor running BASIC. These machines, which are integrated using the Task Control Architecture (TCA), are used for (a) interfacing and central control, (b) map building and position control, (c) planning, (d) vision, (e) speech and (f) fast reactive mechanisms.

NAVIGATION BASED ON SONAR SENSORS

Odysseus' navigation is map-based: the robot progressively constructs a two-dimensional occupancy grid map of its environment based on sonar measurements. Each vector of sonar values is translated to occupancy estimates by a pair of backpropagation networks: One network is trained to map sonar values to probabilities of occupancy, and is used as an inverse model of the sonar sensors. A second network computes the confidence in these estimates, which are used for combining multiple sonar readings. These networks have been trained before the competition to match the characteristics of Odysseus' sonar sensors.

Odysseus uses a fast, anytime A* search algorithm to find a minimal-cost path through free-space, based on the occupancy grid values. The search is significantly accelerated by reusing earlier planning results (partial plans). Additional fast local search is employed to plan for exceptions (e.g. dynamic obstacles, such as walking humans) in advance.

[1] Odysseus was a mythic hero in ancient Greece some 2,500 years ago. Due to some errors in the navigation system (and the anger of the goddess Athena), he spent 10 years wandering through the Mediterranean.

The occupancy map is also used for real-time position estimation. To augment its dead-reckoning, after each step, Odysseus reestimates its position with respect to the global map. This is done by building a local map from the most recent sonar input (using the two artificial neural networks described above), and then optimizing the match between this local map and the global map using gradient descent search. We have found that this procedure successfully deals with cumulative errors in dead-reckoning.

Using the TCA, map building, planning and position estimation is all done in real-time, concurrently with plan execution.

OBSTACLE DETECTION AND COLLISION AVOIDANCE

Odysseus uses two techniques for obstacle detection and avoidance. While the robot moves, it continually monitors its sonar sensor, and slows down and/or stops if something blocks the way. The robot also monitors its wheels encoders to detect stuck or slipping wheels. This "guarded move" is implemented on the robot platform itself, to make it more reactive and independent of radio link delays. In addition, a fast closed-loop controller is employed to navigate around unexpected obstacles and avoid collisions by modifying the direction of the robot. In the current implementation, this controller is realized using a backpropagation network trained with a reinforcement learning algorithm.

VISION

Vision is used primarily for finding and identifying distinguished objects. The Objects are marked by a circular pattern, divided vertically into two regions: (a) a *"checkerboard" region* of black and white squares, which is used to distinguish the object, and (b) an *individual name region*, which consists of horizontal lines that uniquely identify that object.

The camera system periodically takes pictures and looks for objects in the scene. The information extracted from the picture (range, relative direction, identification) is used to construct a two-dimensional target map similar to the occupancy maps built from sonar readings. The map keeps track of the position of detected marked objects as well as known object-free regions in the operation area.

EXPLORATION

Exploration is the process of maximizing knowledge gain. Odysseus evaluates the expected knowledge gain along a path using inverse models of sonar and camera sensors. Odysseus' exploration control consists of two components: (a) Exploration is combined with general navigation in a way that optimizes knowledge gain while the robot is in motion. Consequently, the path planner can take the expected knowledge gain into account. (b) In addition, the path planner can be used for *pure exploration*, operating without a specific goal position. Then the robot navigates to places where the expected (visual) knowledge gain is optimal. Pure exploration is employed to hunt for objects.

In the current implementation, exploitation dominates exploration: Whenever Odysseus knows about some objects which it has not yet been to, it moves towards the nearest one. Here exploration plays a minor role. Once all detected objects have been approached, the robot explores to find new objects.

USER INTERFACE

Odysseus' user interface handles oral and graphical commands.

A speech recognition and generation system is used to command and, if necessary, teleoperate the robot. The Sphinx speech recognition, developed at CMU, performs speaker-independent speech recognition in real-time on arbitrary pre-defined grammars. Oral speech commands include low-level ones such as "stop", as well as high-level commands for specifying tasks, such as "go to object 3, then explore, go to object 15, and return." Odysseus has a speech synthesizer on board, which allows the robot to give feedback and carry on a dialogue with humans.

Odysseus also has a graphical interface, which is used mainly for monitoring the current state of the robot (its maps and plans) as well as the state of the control programs. Odysseus can also be controlled and teleoperated with this graphical interface, although the system usually operates totally autonomously.