Proceedings of the 1991 IEEE
International Conference on Robotics and Automation
Sacramento, California - April 1991

Learning Conditional Effects of Actions for Robot Navigation

Michael Barbehenn and Seth Hutchinson

Artificial Intelligence Research Group
Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana–Champaign
405 North Mathews Ave.
Urbana, IL 61801

Abstract

In this paper we describe GINKO, an integrated learning and planning system that we have applied to an autonomous mobile robot domain. The goal of GINKO's learning system is to partition the robot's configuration space into regions in which actions exhibit a uniform qualitative behavior. This partitioning is performed by an inductive learning algorithm that classifies regions of the configuration space with regard to the effects of the robot's actions when executed in those regions. GINKO's learning is driven by its attempts to perform tasks. Thus, the learned effects of actions are directly applicable to normal system performance.

1. INTRODUCTION

Many of the potential applications for autonomous mobile robots require the robot to function with little or no *a priori* knowledge about its environment. This is true for space exploration, undersea applications, and some industrial settings. To perform effectively in such situations, a robot must be able to adapt to its environment in ways not necessarily envisioned by its designers. To do this, the robot must observe its environment (through external sensors), and ascertain how environmental features affect its performance. This knowledge must then be applied to new problems as they arise.

In this paper, we introduce GINKO, a system that we believe makes a significant step toward this goal. GINKO combines techniques from machine learning with a task planner to learn how to function effectively. Equipped with an initial description of the robot's capabilities, GINKO is instructed to perform various tasks. This initial description takes the form of a set of characteristic regions for the actions, where a characteristic region is a region in the robot's configuration space over which the action exhibits uniform qualitative behavior.

Of course we do not expect that GINKO's planner will be able to produce plans for all tasks, since this would assume that the system designer anticipated all possible contingencies. When the planner fails, GINKO is in a position to learn new ways to achieve goals. There are two ways that the necessary knowledge can be acquired: GINKO can ask for assistance from a human, or it can independently explore possible solutions to the problem. In either case, GINKO is able to learn the effects of actions when they are executed in different regions of configuration space. Thus, the learning component is forced to continually refine and improve the set of characteristic regions for each action as novel configurations are encountered.

To illustrate the utility of GINKO's learning mechanism, in the experiments reported in this paper we have purposely provided the planner with an inadequate initial characterization of the effects of the robot's actions. For example, GINKO is not told that forward progress is impeded when an obstacle is encountered. As our experiments show, such incomplete knowledge causes a number of failures, which are subsequently generalized into a set of characteristic regions for the actions. The qualitative behavior associated with these characteristic regions is "no change in forward position."

The remainder of the paper is organized as follows. We first give an overview of GINKO, a list of its components, and how they interrelate. We then proceed to develop the machine learning background necessary to appreciate the inner workings of the system, its methodology and tradeoffs. Finally we discuss related research and present some preliminary results of this research effort.

2. OVERVIEW

GINKO (Guided Induction and Knowledge Organization) is a system for problem solving in domains where sensor data is continuously available and sufficient for proper monitoring of actions. It is currently being developed on the test–bed of mobile robotics. We assume, for the moment, that the robot is the sole agent in a deterministic world; nothing changes unless the robot effects such a change, with the exception of time. Learning how to navigate is the main focus in this paper. Effective navigation requires planning, monitoring, and dynamic response; these are the subjects discussed herein.

GINKO consists of four main components. It has a set of sensors that continuously monitor the environment. It has a memory which is conceptually divided into two parts: a set of characteristic regions for each action and a description of the associated behaviors (i.e. the description of how the actions behave when executed in various regions of the configuration space), and the sensor data that has been collected for each characteristic region. The performance element, which includes a planner and plan monitors, is responsible for carrying out tasks in the environment. Finally there is an induction component which, upon receiving a description of an action's behavior from the performance element, retrieves relevant data points (previous sensor readings) and generalizes them to construct a new characteristic region for the action. This new characteristic region can be used in future planning efforts.

2.1. CONFIGURATION SPACE

In most robotics literature, a configuration is a geometric description of the position and orientation of an object. Our use of the word is a bit more general. Specifically, by a configuration we mean a complete specification of the state of the robot. Therefore, in GINKO, since the set of values returned by the robot's sensors define the state of the world, a configuration is specified by a tuple of sensor values.

In order to plan a path from an initial configuration to a goal configuration, GINKO must understand the effects of taking each action in every intermediate configuration. In reality, an action cannot be expected to have identical effects in any two distinct configurations, or even the same configuration at distinct points in time. For example, attempting to move forward in a unobstructed situation moves the robot forward, but by varying amounts; there might be loose dirt or oil on the surface or some slippage in a gear meshing. In order to create successful plans under such circumstances, the planner would require a complete characterization of the effects of each action in every individual configuration in the configuration space. Planning with such exactness is intractable; and storing that much information exceeds the capabilities of today's computer systems.

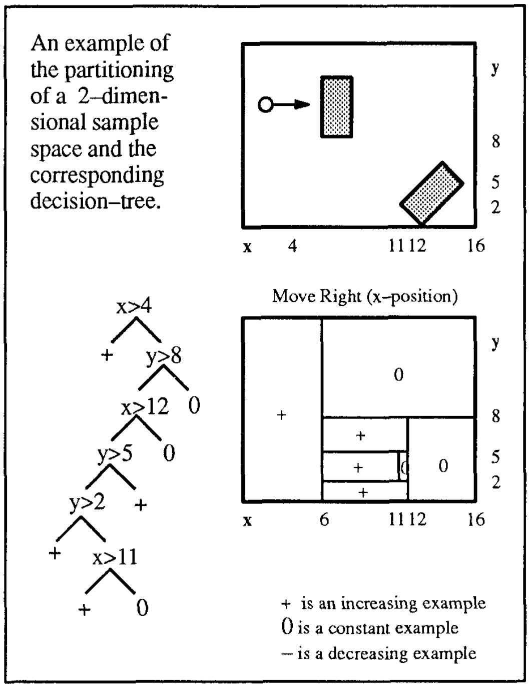
By assuming that executing actions in neighboring configurations results in similar behaviors, a great many configurations can be described by one contiguous characteristic region of configuration–space. By a behavior, we mean the aggregate qualitative (monotonic) changes observed in the sensor values. These qualitative changes are classified as being either increasing, decreasing, or constant. More precisely, the sign of the difference between every consecutive pair of sensor readings is a qualitative vector of change, and this change is referred to as the observed behavior. In our formulation, we assume that there are no temporally distal effects, so that the behavior observed in each configuration can be attributed to the just–executed action.

These assumptions give GINKO license to generalize beyond what it has observed. *Each generalization is a hypothesis about the behavior of an action when it is executed in a particular characteristic region of the configuration space.* In part, the size and shape of these hypothesized regions are functions of the representation language used by the induction component. Here, each region corresponds to a conjunction of sensor value ranges. In general, there are several disjoint characteristic regions associated with a particular qualitative behavior of an action [Rendell90].

2.2. INDUCTION

Induction, in machine learning, is a generalization process by which a set of examples is partitioned into concept classes and a description of each class is obtained. (For this paper, a concept class corresponds to a set of characteristic regions in configuration–space for which an action exhibits uniform qualitative behavior.) One of the most difficult problems faced by an inductive learning system is that of accurately partitioning the instance space (or, in our case , the configuration space). Optimally, characteristic regions will be such that the action will exhibit the same qualitative behavior when executed at any configuration in the region. This might be relaxed depending on the amount of noise in the sensor values, the description language used, the desired complexity of the description, and the desired accuracy.

Consider the example illustrated by the figure below. In this example, the mobile robot is placed in a room containing two obstacles, both of which are rectangular. The robot is told to execute the action that moves it to the right in the room. At first, the robot is able to move freely to the right (sample data points along this successful path are illustrated by the symbol '+'). After a short while, the



An example of the partitioning of a 2–dimensional sample space and the corresponding decision–tree.

first obstacle is encountered, and forward progress is halted (these points are illustrated by the symbol '0'). At this point, a partition is erected that divides the space into two regions, one in which the action has the effect of moving the robot to the right, and the other in which the action has no effect on the position of the robot. As the robot continues roaming about, it will encounter more instances of both success and failure, and in the process, new partitions will added. The figure illustrates how a number of these partitions will be constructed. Note that all partitions are orthogonal to an axis of the configuration space.

The induction algorithm used by GINKO is a variant of Rendell's PLS. (For example, our version uses data–directed trial splits rather than fixed increments and has been generalized to handle multiple classes.) PLS is appropriate because it was designed for numeric data and it has been shown to perform well [Rendell89]. PLS takes as input a list of classified examples and returns a partition of instance space that separates examples of different class membership. The examples in our system are configurations that are classified by whether or not executing a specified action had a desired effect.

The action behaviors are analyzed one dimension at a time (recall that a configuration is defined by a tuple of sensor values, and therefore each sensor defines one dimension in the configuration space). Trial splits are examined to see how well they separate regions of the configuration space based on uniformity of action effects. A good split produces two regions of high purity (the configurations are classified similarly) to replace one with lower purity. The best split for each dimension is kept. When all dimensions have been analyzed, the overall single best split is erected and the program calls itself recursively on each half. The resultant regions (which are hypercubes, since all splits are orthogonal to axes of the configuration space) can be represented in a decision tree with their associated data.

The PLS metric for measuring the goodness of a trial split, called the dissimilarity, is defined by the function $d = |\log u_1 - \log u_2| - t_\alpha \log e_1 e_2$ where u_i is a class membership probability estimated by the relative proportion of positive configurations in the region, e_i the error factor due to finite sample size, and t_α the confidence factor [Rendell86]. The dissimilarity metric measures the difference in the relative proportions of positive examples in the two regions.

Trial splits may be made in various ways. This is one source of variation in the partitioning of configuration–space. When separating configurations of differing class membership, there is great freedom in where the boundary is placed. A conservative, or pessimistic, algorithm places the boundary so that it just includes the positive configuration—thus by default, all unknown configurations are negative. A liberal, or optimistic, algorithm places the boundary so that it just excludes the negative configuration—thus by default, all unknown configurations are positive.

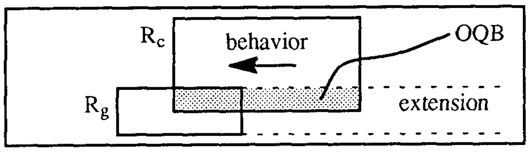
2.3. PERFORMANCE ELEMENT

The plans generated by GINKO consist of sequences of monitored actions. The monitors can be either simple predicates that monitor sensor values to determine when halting criteria have been achieved, or reactive monitors that monitor for error conditions and construct local recovery plans.

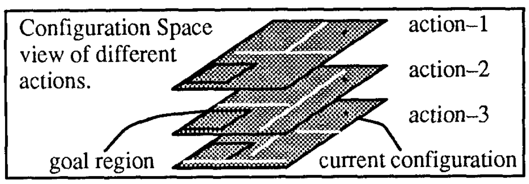
As new data is acquired (by monitoring the sensors during plan execution), it is associated with the characteristic regions of the configuration space, as generated by PLS. When a datum is misclassified (i.e. associated with a characteristic region which has a different associated behavior) the characteristic region is marked as incorrect and is eventually repartitioned. Two parameters control the repartitioning process. The strategy for repartitioning is either complete (all of configuration space) or incremental (only those regions which are incorrect). The strategy for consistency checking is either active (errors are handled immediately) or passive (error correction is deferred).

3. CONFIGURATION SPACE PLANNING

The planning technique developed for GINKO is termed Orthogonal–Qualitative Backprojection (OQB). The definition of an OQB is best introduced by an example. Consider the figure shown below. Here, the robot's task is to move into the indicated region, R_g. In order to accomplish this, the planner searches its possible actions for any action that has an associated qualitative behavior that would move the robot into the goal region. The figure shows one such characteristic region, R_c (the left pointing arrow indicates that the direction of qualitative change is to the left for the particular action when executed from within R_c). Now, notice that the robot will only move into the goal region from this characteristic region if it is in the shaded sub–region. This shaded sub–region is one OQB for the goal region. More precisely, the OQB of a particular goal region R_g with respect to a characteristic region R_c is the intersection of R_c with the extension of R_g in the inverse direction of the qualitative change associated with R_c. Intuitively, an OQB is a region within which executing the associated action will head the robot towards the goal. We note that if the direction of qualitative change for R_c had been down and to the left, then the entire region R_c would be included in the OQB.



The search for a plan is characterized by a subgoal region (initially the goal region), a function that returns the OQBs of the subgoal region for each action, and a stopping criterion. Pictured below are the roles of three actions with their partitions of configuration space for the given planning problem.



Given a solution sequence of OQBs and associated actions, the final generated plan is the sequence of actions with associated monitors, which are either simple or reactive. The basic monitor checks whether the current configuration is in the goal region (to stop under fortuitous events), whether the next region has been entered, and finally, whether the current OQB has been prematurely exited. In this latter case, the reactive monitor inserts a patch plan to get from the current situation to the next OQB in the original sequence. This is a natural solution to the problem of ambiguity native to qualitative analysis.

This view of planning is an integration [Gervasio90] of classical [Chapman87, Hutchinson90] and reactive [Agre87, Schoppers87] paradigms. It is classical in that regions are chained together to form explanations or proofs of success. It is reactive in that, for the most part, the exactness and optimality of the plans has been replaced with a strong notion of progress.

4. RELATED RESEARCH

Research related to that presented in this paper falls into a number of broad categories: learning for autonomous navigation in mobile robotics, planning, and inductive machine learning. In this section we briefly review relevant literature from each of these categories.

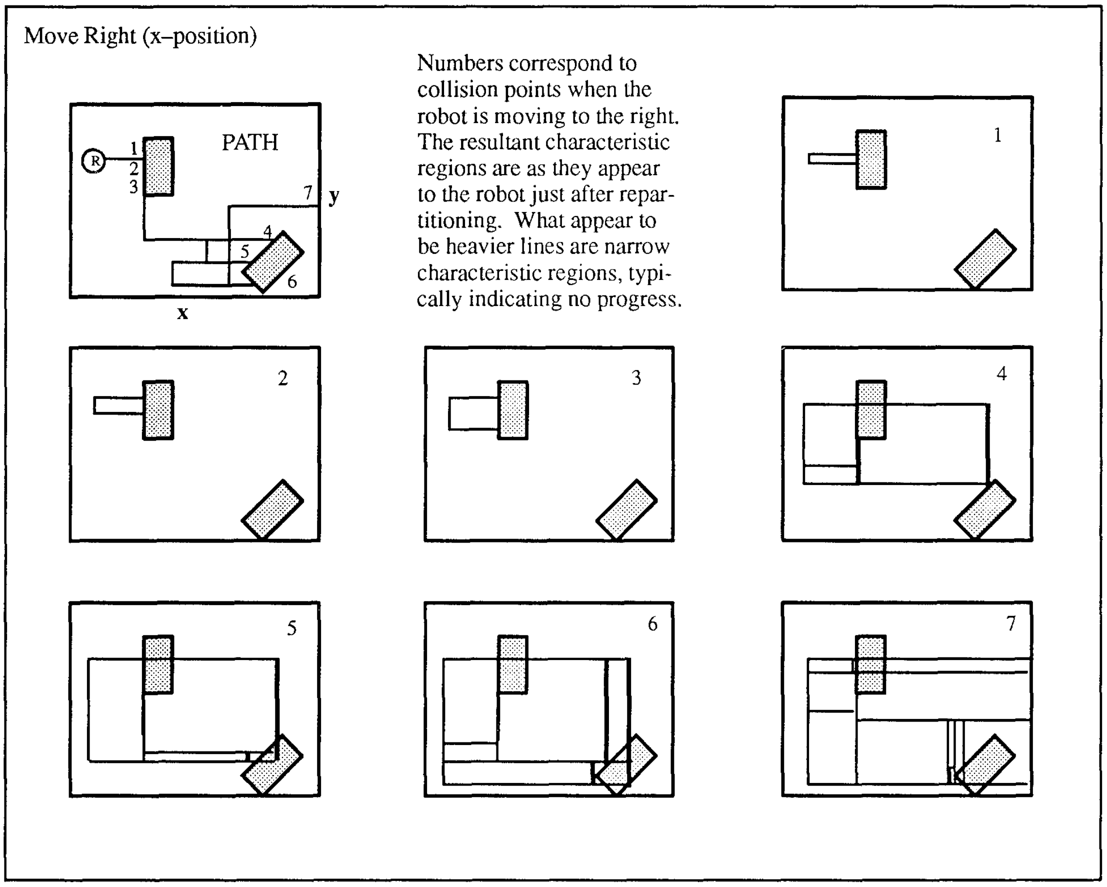
Most previous research in learning for autonomous navigation has focussed on mapping out previously unexplored terrains. An algorithm to construct symbolic descriptions of an unknown environment is described in [Oommen87]. This algorithm constructs a visibility graph of the environment by performing random traversals in the environment. A systematic approach to the same problem is described in [Rao88], in which the robot visits the vertices in the visibility graph in a depth–first manner. Elfes describes an approach to building multilevel descriptions of the robot's environment by fusing range information obtained by a sonar range finder from multiple view points [Elfes87]. Ayache and Faugeras take a different approach to the same problem [Ayache88, Ayache89]. In their approach, a passive vision system is used to obtain 3–D data, which is fused using an Extended Kalman Filter. This yields a geometric description of the environment, which includes geometric uncertainty. In addition to these recent efforts, additional research in the area of mapping unexplored terrains can be found in [Brooks85, Chattergy85, Crowley85, Laumond83, Turchen85].

Gervasio's method of integrating reactivity into a classical planner is closest to GINKO's planning method [Gervasio90]. Her planner defers reasoning about goals which are known to be achievable at plan time; instead it monitors such actions at run–time to take advantage of the current situation. Furthermore, the analysis is at the qualitative level.

Related machine learning research includes systems which combine empirical with analytical learning [Danyluk89, Pazzani88, Rajamoney87]. These systems emphasize the use of domain theories to constrain generalization of planning traces.

5. RESULTS AND DISCUSSION

An experiment was carried out to shed light on the nature and direction of this research. In this first experiment, a robot with two–degrees of freedom was set to wander randomly. Pictured below are the configuration space



Numbers correspond to collision points when the robot is moving to the right. The resultant characteristic regions are as they appear to the robot just after repartitioning. What appear to be heavier lines are narrow characteristic regions, typically indicating no progress.

partitions corresponding to the x–position sensor and the move–right action for the first 200 time steps of the experiment. The outermost rectangle shown depicts the sensor extremes as known to the robot. The actual regions extend into infinity at the edges of this rectangle.

The sequence illustrates the basic functioning of GINKO as it confronts novel configurations. One of the interesting results of this experiment, which ran for more than 20,000 time steps, was that the frequency of repartitioning, and hence misclassification rate, appeared to be exponentially decreasing. At the same time, the number of characteristic regions grew considerably.

Note that planning with faulty partitions leads to errors but, more fundamentally, the planner cannot guarantee that travel through an OQB can be maintained. This inherent ambiguity arises from the qualitative analysis of the data. Reincorporating some quantitative information might remedy this small drawback.

Our aim has been to acquaint the reader with our learning architecture and indicate some of the many areas for future research. Among these include more intelligent control over the splitting criterion in PLS, and hybrid strategies to control when and how repartitioning occurs.

REFERENCES

[Agre87] P. Agre and D. Chapman, "Pengi: An Implementation of a Theory of Activity," *Proceedings of the National Conference on Artificial Intelligence*, Seattle, WA, July 1987, pp. 268–272.

[Ayache88] N. Ayache and O. D. Faugeras, "Building, Registrating, and Fusing Noisy Visual Maps," *Int'l Journal of Robotics Research 7*, 6 (December 1988), pp. 45–65.

[Ayache89] N. Ayache and O. D. Faugeras, "Maintaining Representations of the Environment of a Mobile Robot," *Institute of Electrical and Electronics Engineers Journal of Robotics and Automation 5*, 6 (December 1989), pp. 804–819.

[Brooks85] R. A. Brooks, "Visual Map Making for a Mobile Robot," *Proc. Institute of Electrical and Electronics Engineers Conf. on Robotics and Automation*, 1985, pp. 824–829.

[Chapman87] D. Chapman, "Planning for Conjunctive Goals," *Artificial Intelligence 32*, 3 (1987), pp. 333–378.

[Chattergy85] R. Chattergy, "Some Heuristics for the Navigation of a Robot," *International Journal of Robotics Research 4*, 1 (Spring 1985), pp. 59–66.

[Crowley85] J. L. Crowley, "Navigation of an Intelligent Mobile Robot," *Institute of Electrical and Electronics Engineers Journal of Robotics and Automation RA–1*, 2 (March 1985), pp. 31–41.

[Danyluk89] A. P. Danyluk, "Finding New Rules for Incomplete Theories: Explicit Biases for Induction with Contextual Information," *Proceedings of the Sixth International Workshop on Machine Learning*, Ithaca, NY, June 1989, pp. 34–36.

[Elfes87] A. Elfes, "Sonar–Based Real–World Mapping and Navigation," *Institute of Electrical and Electronics Engineers Journal of Robotics and Automation RA–3*, 3 (June 1987), pp. 249–265.

[Gervasio90] M. T. Gervasio, "Learning General Completable Reactive Plans," *Proc. Eighth American Association for Artificial Intelligence*, Boston, MA, August 1990, pp. 1016–1021.

[Hutchinson90] S. A. Hutchinson and A. C. Kak, "SPAR: A Planner that Satisfies Operational and Geometric Goals in Uncertain Environments," *Artificial Intelligence Magazine 2*, 1 (Spring 1990), pp. 30–61.

[Laumond83] J. Laumond, "Model Structuring and Concept Recognition: Two Aspects of Learning for a Mobile Robot," *Proceedings of the National Conference on Artificial Intelligence*, Karlsruhe, W. Germany, August 1983.

[Oommen87] B. J. Oommen, S. Iyengar, N. S. V. Rao and R. L. Kashyap, "Robot Navigation in Unknown Terrains Using Learned Visibility Graphs. Part I: The Disjoint Convex Obstacle Case," *Institute of Electrical and Electronics Engineers Journal of Robotics and Automation RA–3*, 6 (December 1987), pp. 672–681.

[Pazzani88] M. J. Pazzani, "Integrated Learning with Incorrect and Incomplete Theories," *Proc. Fifth International Conf. on Machine Learning*, Ann Arbor, MI, July 1988, pp. 291–297.

[Rajamoney87] S. Rajamoney and G. DeJong, "The Classification, Detection and Handling of Imperfect Theory Problems," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 205–207.

[Rao88] N. S. V. Rao, S. Iyengar, B. J. Oommen and R. L. Kashyap, "On Terrain Model Acquisition by a Point Robot Amidst Polyhedral Obstacles," *Institute of Electrical and Electronics Engineers Journal of Robotics and Automation 4*, 4 (August 1988), pp. 450–455.

[Rendell86] L. Rendell, "A General Framework for Induction and a Study of Selective Induction," *Machine Learning 1*, 2 (1986), pp. 177–226.

[Rendell89] L. Rendell, H. Cho and R. Seshu, "Improving the Design of Similarity–Based Rule–Learning Systems," *International Journal of Expert Systems 2*, 1 (1989), pp. 97–133.

[Rendell90] L. Rendell and H. Cho, "Empirical Learning as a Function of Concept Character," *Machine Learning 5*, 3 (August 1990), pp. 267–298.

[Schoppers87] M. J. Schoppers, "Universal Plans for Reactive Robots in Unpredictable Environments," *Proceedings of the Tenth International Joint Conference on Artificial Intelligence*, Milan, Italy, August 1987, pp. 1039–1046.

[Turchen85] M. P. Turchen and A. K. C. Wong, "Low Level Learning for a Mobile Robot: Environmental Model Acquisition," *Proc. Second Int'l Conf. on Artificial Intelligence and Its Applications*, December 1985, pp. 156–161.