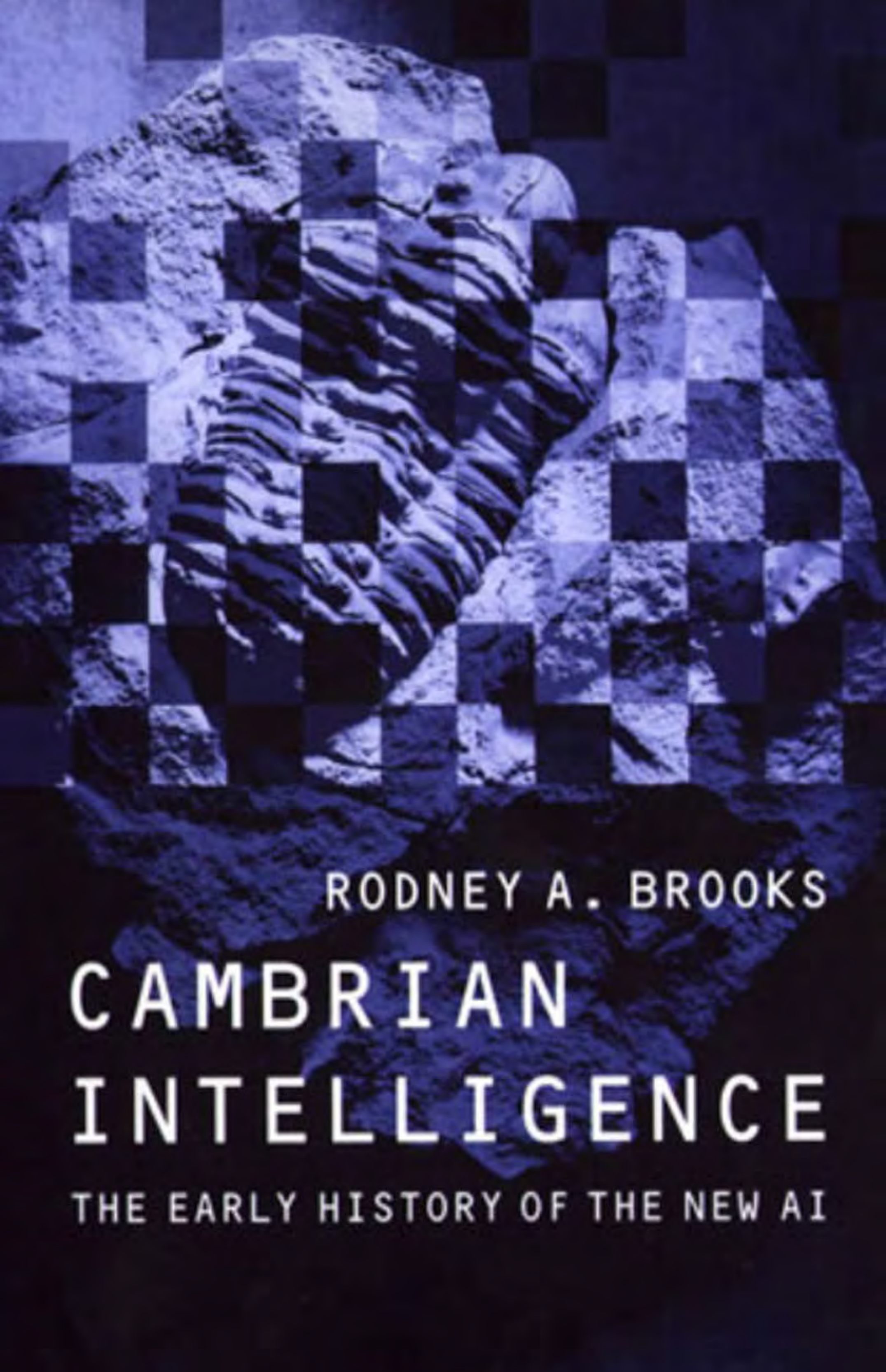

RODNEY A. BROOKS

CAMBRIAN INTELLIGENCE

THE EARLY HISTORY OF THE NEW AI

CAMBRIAN INTELLIGENCE

CAMBRIAN INTELLIGENCE

The Early History of the New AI

RODNEY A. BROOKS

A Bradford Book
The MIT Press
Cambridge, Massachusetts
London, England

This book was set in Computer Modern by the author using $\LaTeX$ and was printed and bound in the United States of America.

Library of Congress Cataloging-in-Publication Data

Brooks, Rodney Allen.
Cambrian intelligence: the early history of the new AI / Rodney A. Brooks.
p. cm.
"A Bradford Book."
Includes bibliographical references.
ISBN 0-262-02468-3 (hc.: alk. paper)–0-262-52263-2 (pbk.: alk. paper)
1. Robotics. 2. Artificial intelligence. I. Title.
TJ211.B695 1999
629.8'9263–dc21

99-17220
CIP

Contents

Preface

This is a collection of scientific papers and essays that I wrote from 1985 to 1991, a period when the key ideas of behavior-based robotics were developed. Behavior-based robotics has now been widely adopted for mobile robots of all sorts on both Earth and Mars.

Mobile robotics was an almost non-existent curiosity fourteen years ago. There were many unmanned guided vehicles developed to carry parts in automated manufacturing plants, but they just had to follow guide wires in the ground. In contrast, autonomous mobile robots are conceived of as robots that can operate in worlds where the layout of obstacles are not known ahead of time; they are to operate in the sorts of worlds that people and animals can negotiate with ease. The few autonomous mobile robots that did exist fourteen years ago were programmed to build elaborate three-dimensional world models and to *plan* a long series of actions within those world models before they started to move. Out of necessity the handful of researchers in the area made a restriction that their robots could operate only in static worlds.

The behavior-based approach to mobile robots changed the way robots were programmed, at least at the lower levels of control. The eight chapters of this book are papers that I wrote (one with a student) introducing the behavior-based approach; four are technical papers describing robot systems and approaches, and four are philosophical papers describing the changes in intellectual point of view that enables this approach.

In an idealized scientific world one might think that the new intellectual point of view was developed first and then the technology and technicians caught up and implemented systems guided by those new intellectual points of view. But that is certainly not the way things developed for me. In all cases the technological implementations came first, and the philosophical realizations followed later. For this reason the technical papers are presented first and the philosophical papers second. However, either part of the book can be read first as all the papers were written to be read as single works. Philosophers may safely

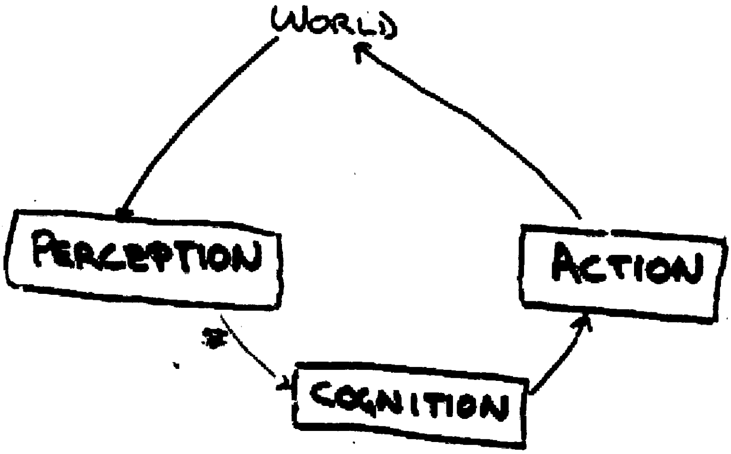

Figure 1: The traditional model where cognition mediates between perceptions and plans of actions.

skip the technical papers without finding themselves ungrounded in the philosophical sections.

While technology definitely led philosophy there was one key earlier realization, philosophical in nature, that was critical to the development of the behavior-based approach. In hindsight, it may seem to be the critical realization that enabled and outlines the whole approach but I must say that the new approach was not at all obvious at the time of the realization.

The realization was that the so-called central systems of intelligence — or *core AI* as it has been referred to more recently — was perhaps an unnecessary illusion, and that all the power of intelligence arose from the coupling of perception and actuation systems. This is the cornerstone of behavior-based robotics, both mobile robots as they have developed over the last twelve years and humanoid robots that have been developed more recently.

For me this realization came about after spending a number of years working on both perception, mostly vision-based perception, and motion planning, mostly for abstract two-dimensional shapes and some for robot manipulator arms.

I had always assumed that there would be some intelligent system coupling the two of them together doing so-called *high-level reasoning.* Indeed this was the generally accepted order of things, largely validated by the early successes of the Shakey robot at the SRI Artificial Intelligence Center in Menlo Park, California, near Stanford University. The higher-level reasoning system of Shakey worked in predicate logic, and the perception system produced descriptions of the world in first-order predicate calculus. However, that work in the late sixties and very early seventies was carried out in a very restricted world; a well-lit area consisting of only matte-painted walls, clean floors, and large cubes and wedges with each face painted a different color. These restrictions made the vision problem fairly straightforward though not trivial.

By the early eighties it seemed to me that if there was to be any validity to this approach then computer vision should be able to deliver predicate calculus descriptions of much less constrained and much more complex worlds. Computer vision systems ought to be able to operate in the ordinary sorts of environments that people operated in, cluttered offices with things stuck on walls and disorderly piles of papers that partially obscured objects. A computer monitor, for instance, should be visually recognizable as such even if it were a new model with a shape and size slightly different from all those that the vision system had been told about or had seen before. Even more challenging, a computer vision system should be able to operate outdoors and pick out trees, hills, pathways, curbs, houses, cars, trucks, and everything else that a three-year-old child could name.

There were no vision systems around doing anything even remotely as sophisticated. Any such dreams had been put on hold by the computer vision community as they worked on very difficult but much simpler challenges. But even so, the validity of the approach was not questioned. Rather, the progress in computer vision was questioned. Alex (Sandy) Pentland, then a young researcher at SRI held a weekly seminar series to try to uncover the sources of the seeming stall in computer vision. The title of the seminar series was *From Pixels to Predicates*, setting the tone for the direction of inquiry. I was a very junior faculty member at Stanford at the time and was invited to give one of the weekly talks during the 1983–84 academic year.

I have kept my crude hand-drawn opening slides from that talk, and they are reproduced in figures 1 and 2. I did not have a cogent argument to make based on these slides. It may have been that I had recently discovered the joy of brashly telling everyone that their most implicit beliefs or assumptions were wrong, or at least open to question; the central systems people were giving us perception people a hard time that we just weren't producing the sorts of outputs that would enable their *intelligent* systems to do all the really hard stuff. Or it may have been that I was extraordinarily frustrated both with the difficulty of producing complete descriptions of the world from visual input, and with the immense complexity that had to be maintained in such models of the world in order to plan successful motor actions within them. Or I may have had an incredibly clever insight. I doubt the latter, and think it was a combination of the first two, perhaps dominated by the perverse pleasure of the first one.

In any case there were actually three transparencies. An underlying one and two overlays that gave rise to the two different figures. The underlying transparency had the words PERCEPTION, WORLD, and ACTION written on them, with two arrows, one from action to the world, and one from the world to perception, indicating the causality chain of the model. The first overlay gave the image reproduced in figure 1. In this, COGNITION was the thing, the computational box, that was used to decipher perceptions and to order up actions. This was very much the common implicitly held view of how to build an intelligent system, and was certainly the approach implicitly promoted by Sandy's seminar. The basic idea is that perception goes on by itself, autonomously producing world descriptions that are fed to a cognition box that does all the real *thinking* and instantiates the real *intelligence* of the system. The thinking box then tells the action box what to do, in some sort of high-level action description language.

In my talk I then changed the overlay to give the image in figure 2. This completely turns the old approach to intelligence upside down. It denies that there is even a box that is devoted to cognitive tasks. Instead it posits both that the perception and action subsystems do all the work and that it is only an external observer that has anything to do with cognition, by way of attributing cognitive abilities to a system that works well in the world but has no explicit place where cognition is done.

The first model has the recursive problem of deciding what is in the cognition box, and perhaps finding yet another little homunculus inside there until eventually all the power of reason is reduced to a rather dumb little computational intelligence operating in such highly abstract terms that there is no real place for truly intelligent action to be generated. Rather, it is slightly dressed up and made more real as each level of recursion is unwound.

The second model solves this whole problem by denying its existence. Of course, I had absolutely no idea at the time how to blend the perceptual and actuator systems to work together and to achieve all the desired behaviors. And there was certainly no hint of it in the actual paper I contributed to an edited book on the series that Sandy produced.

In retrospect, my students and I have spent the last fourteen years trying to justify that second figure, by building ever more complex artifacts which demonstrate intelligent behavior using the second model.

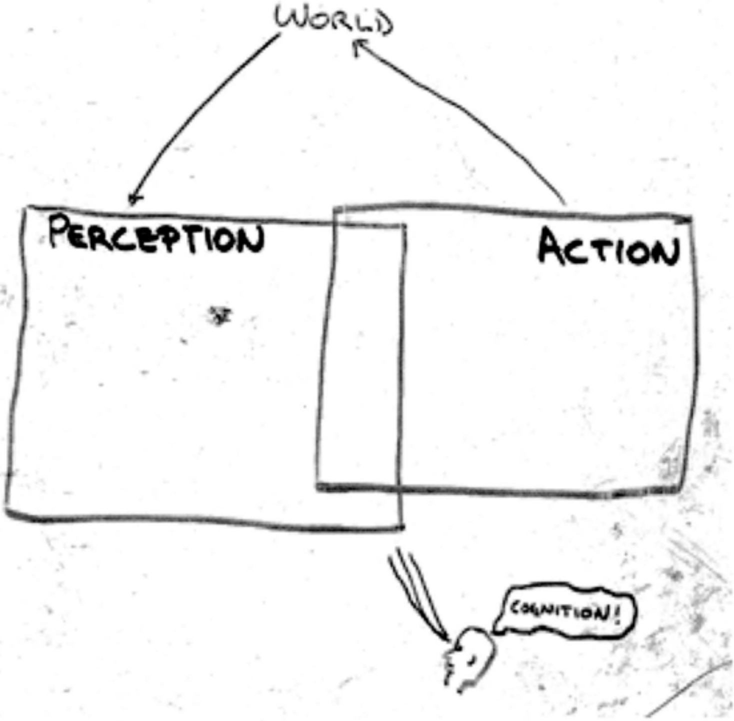

Figure 2: The new model, where the perceptual and action subsystems are all there really is. Cognition is only in the eye of an observer.

The eight papers in this book represent my contributions to the development of the behavior-based approach to robotics. This approach has now achieved success across two planets. Hundreds of people are developing new applications using the approach; applications in entertainment, service industries, agriculture, mining, and the home. There is a real chance that these developments will soon bring about yet another change in humanity's relationship with machines following our earlier agricultural, industrial, and information revolutions: the robot revolution.

Acknowledgments

I have had many students over the years who have contributed to the formation of my ideas and helped with the technological implementations. Those who worked with me during the period that this book covers in-

clude Phil Agre, Colin Angle, Cynthia Breazeal (Ferrell), David Chapman, Mike Ciholas, Jon Connell, Peter Cudhea, Anita Flynn, Chris Foley, Ian Horswill, Peter Ning, Pattie Maes, Maja Mataric, Lynne Parker, and Paul Viola.

A number of my colleagues at the MIT Artificial Intelligence Laboratory have been useful sounding boards, and sometimes I have even convinced them of my arguments. Those that I harassed during the times of these papers include Eric Grimson, Tomás Lozano-Pérez, David Kirsh, and Thomas Marill.

There were other colleagues who directly helped me with these ideas, including Leslie Kaelbling, Grinnell More, and Stan Rosenschein.

Annika Pfluger has assisted me daily for many years, and she was very active in bringing this volume to fruition, both with the references and the figures.

Cambridge, Massachussets
March, 1999

PART I

……

TECHNOLOGY

CHAPTER 1

A ROBUST LAYERED CONTROL SYSTEM FOR A MOBILE ROBOT

This is by far the most referenced paper that I have written. It both introduced the notion of behavior-based robotics, although that name only came much later, and described the first instance of a robot programmed in this manner. I first gave a talk on this work in Gouvieux-Chantilly, France, in late 1985. Only many years later did I learn that in the back row senior robotics people were shaking their heads asking each other why I was throwing my career away. The content of this paper was shocking because it argued for simplicity rather than for mathematical complexity of analysis and implementation. To this day many people still find the contents of this paper entirely disreputable because it is not filled with mathematical equations. Having spent six years in a mathematics department I am not afraid of mathematics, indeed I revere beautiful mathematics. But I am afraid that many people have severe cases of physics envy *and feel that their work is not complete if it does not have pages of equations, independently of whether those equations shed any light at all on the deep questions. To my mind there has to date been no decent mathematical analysis of the ideas presented in this paper and further developed by many, many researchers during the rise of behavior-based robotics. That is not to say there should not or can not be such an analysis. But I think it will require some very deep insights and can not rely on surface level equation generation.*

Abstract. We describe a new architecture for controlling mobile robots. Layers of control system are built to let the robot operate at increasing levels of competence. Layers are made up of asynchronous modules which communicate over low bandwidth channels. Each module is an instance of a fairly simple computational machine. Higher level layers can subsume the roles of lower levels by suppressing their outputs. However,

---

Originally appeared in the *IEEE Journal of Robotics and Automation, RA-2* April, 1986, pp. 14–23.

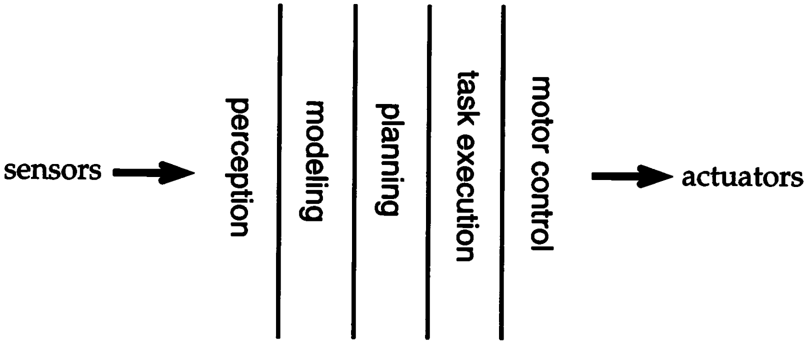

Figure 1: A traditional decomposition of a mobile robot control system into functional modules.

lower levels continue to function as higher levels are added. The result is a robust and flexible robot control system. The system has been used to control a mobile robot wandering around unconstrained laboratory areas and computer machine rooms. Eventually it is intended to control a robot that wanders the office areas of our laboratory, building maps of its surroundings using an onboard arm to perform simple tasks.

1 Introduction

A control system for a completely autonomous mobile robot must perform many complex information processing tasks in real time. It operates in an environment where the boundary conditions (viewing the instantaneous control problem in a classical control theory formulation) are changing rapidly. In fact the determination of those boundary conditions is done over very noisy channels since there is no straightforward mapping between sensors (e.g., TV cameras) and the form required of the boundary conditions.

The usual approach to building control systems for such robots is to decompose the problem into a series (roughly) of *functional units* as illustrated by a series of vertical slices in Figure 1. After analyzing the computational requirements for a mobile robot we have decided to use *task achieving behaviors* as our primary decomposition of the problem. This is illustrated by a series of horizontal slices in Figure 2. As with a functional decomposition we implement each slice explicitly then tie them all together to form a robot control system. Our new decomposition leads to a radically different architecture for mobile robot control systems, with radically different implementation strategies plausible at

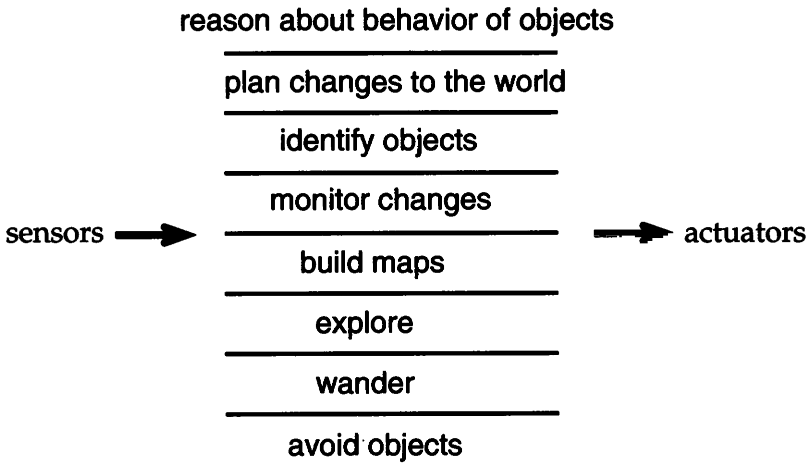

Figure 2: A decomposition of a mobile robot control system based on task achieving behaviors.

the hardware level, and with a large number of advantages concerning robustness, buildability and testability.

1.1 Requirements

We can identify a number of requirements of a control system for an intelligent autonomous mobile robot. They each put constraints on possible control systems we might build and employ.

Multiple Goals: Often the robot will have multiple goals, some conflicting, which it is trying to achieve. It may be trying to reach a certain point ahead of it while avoiding local obstacles. It may be trying to reach a certain place in minimal time while conserving power reserves. Often the relative importance of goals will be context dependent. Getting off the railroad tracks when a train is heard becomes much more important than inspecting the last 10 track ties of the current track section. The control system must be responsive to high priority goals, while still servicing necessary "low level" goals (e.g., in getting off the railroad tracks it is still important that the robot maintains its balance so it doesn't fall down).

Multiple Sensors: The robot will most likely have multiple sensors (e.g., TV cameras, encoders on steering and drive mechanisms, and perhaps infrared beacon detectors, an inertial navigation system, acoustic rangefinders, infrared rangefinders, access to a global positioning satellite system, etc.). All sensors have an error component in their readings.

Furthermore, often there is no direct analytic mapping from sensor values to desired physical quantities. Some of the sensors will overlap in the physical quantities they measure. They will often give inconsistent readings—sometimes due to normal sensor error and sometimes due to the measurement conditions being such that the sensor (and subsequent processing) is being used outside its domain of applicability. Often there will be no analytic characterization of the domain of applicability (e.g., under what precise conditions does the Sobel operator return valid edges?). The robot must make decisions under these conditions.

Robustness: The robot ought to be robust. When some sensors fail it should be able to adapt and cope by relying on those still functional. When the environment changes drastically it should be able to still achieve some modicum of sensible behavior, rather then sit in shock, or wander aimlessly or irrationally around. Ideally it should also continue to function well when there are faults in parts of its processor(s).

Extensibility: As more sensors and capabilities are added to a robot it needs more processing power; otherwise the original capabilities of the robot will be impaired relative to the flow of time.

1.2 Other Approaches

Multiple Goals: Elfes & Talukdar (1983) designed a control language for Moravec (1983)'s robot which tried to accommodate multiple goals. It mainly achieved this by letting the user explicitly code for parallelism and to code an exception path to a special handler for each plausible case of unexpected conditions.

Multiple Sensors: Flynn (1985) explicitly investigated the use of multiple sensors, with complementary characteristics (sonar is wide angle but reasonably accurate in depth, while infrared is very accurate in angular resolution but terrible in depth measurement). Her system has the virtue that if one sensor fails the other still delivers readings that are useful to the higher level processing. Giralt, Chatila & Vaisset (1984) use a laser range finder for map making, sonar sensors for local obstacle detection, and infrared beacons for map calibration. The robot operates in a mode where one particular sensor type is used at a time and the others are completely ignored, even though they may be functional. In the natural world multiple redundant sensors are abundant. For instance Kreithen (1983) reports that pigeons have more than four independent orientation sensing systems (e.g., sun position compared to internal biological clock). It is interesting that the sensors do not seem to be combined but rather, depending on the environmental conditions and operational level of sensor subsystems, the data from one sensor tends to dominate.

Robustness: The above work tries to make systems robust in terms of sensor availability, but little has been done with making either the behavior or the processor of a robot robust.

Extensibility: There are three ways this can be achieved without completely rebuilding the physical control system. (1) Excess processor power which was previously being wasted can be utilized. Clearly this is a bounded resource. (2) The processor(s) can be upgraded to an architecturally compatible but faster system. The original software can continue to run, but now excess capacity will be available and we can proceed as in the first case. (3) More processors can be added to carry the new load. Typically systems builders then get enmeshed in details of how to make all memory uniformly accessible to all processors. Usually the cost of the memory to processor routing system soon comes to dominate the cost (the measure of cost is not important—it can be monetary, silicon area, access time delays, or something else) of the system. As a result there is usually a fairly small upper bound (on the order of hundreds for traditional style processing units; on the order to tens to hundreds of thousands for extremely simple processors) on the number of processors which can be added.

1.3 Starting Assumptions

Our design decisions for our mobile robot are based on nine dogmatic principles (six of these principles were presented more fully in Brooks (1985)):

1. Complex (and useful) behavior need not necessarily be a product of an extremely complex control system. Rather, complex behavior may simply be the reflection of a complex environment (Simon 1969). It may be an observer who ascribes complexity to an organism—not necessarily its designer.

   Things should be simple. This has two applications. (1) When building a system of many parts one must pay attention to the interfaces. If you notice that a particular interface is starting to rival in complexity the components it connects, then either the interface needs to be rethought or the decomposition of the system needs redoing. (2) If a particular component or collection of components solves an unstable or ill-conditioned problem, or, more radically, if its design involved the solution of an unstable or ill-conditioned problem, then it is probably not a good solution from the standpoint of robustness of the system.

3. We want to build cheap robots which can wander around human inhabited space with no human intervention, advice or control and at the same time do useful work. Map making is therefore of crucial importance even when idealized blue prints of an environment are available.

4. The human world is three dimensional; it is not just a two dimensional surface map. The robot must model the world as three dimensional if it is to be allowed to continue cohabitation with humans.

5. Absolute coordinate systems for a robot are the source of large cumulative errors. Relational maps are more useful to a mobile robot. This alters the design space for perception systems.

6. The worlds where mobile robots will do useful work are not constructed of exact simple polyhedra. While polyhedra may be useful models of a realistic world, it is a mistake to build a special world such that the models can be exact. For this reason we will build no artificial environment for our robot.

7. Sonar data, while easy to collect, does not by itself lead to rich descriptions of the world useful for truly intelligent interactions. Visual data is much better for that purpose. Sonar data may be useful for low level interactions such as real time obstacle avoidance.

8. For robustness sake the robot must be able to perform when one or more of its sensors fails or starts giving erroneous readings. Recovery should be quick. This implies that built-in self calibration must be occurring at all times. If it is good enough to achieve our goals then it will necessarily be good enough to eliminate the need for external calibration steps. To force the issue we do not incorporate any explicit calibration steps for our robot. Rather we try to make all processing steps self calibrating.

9. We are interested in building *artificial beings*—robots that can survive for days, weeks and months, without human assistance, in a dynamic complex environment. Such robots must be self sustaining.

2 Levels and Layers

There are many possible approaches to building an autonomous intelligent mobile robot. As with most engineering problems they all start by decomposing the problem into pieces, solving the subproblems for each piece, and then composing the solutions. We think we have done the first of these three steps differently to other groups. The second and third steps also differ as a consequence.

2.1 Levels of Competence

Typically mobile robot builders (e.g., (Nilsson 1984), (Moravec 1983), (Giralt et al. 1984), (Kanayama 1983), (Tsuji 1985), (Crowley 1985)) have sliced the problem into some subset of:

- sensing,
- mapping sensor data into a world representation,
- planning,
- task execution, and
- motor control.

This decomposition can be regarded as a horizontal decomposition of the problem into vertical slices. The slices form a chain through which information flows from the robot's environment, via sensing, through the robot and back to the environment, via action, closing the feedback loop (of course most implementations of the above subproblems include internal feedback loops also). An instance of each piece must be built in order to run the robot at all. Later changes to a particular piece (to improve it or extend its functionality) must either be done in such a way that the interfaces to adjacent pieces do not change, or the effects of the change must be propagated to neighboring pieces, changing their functionality too.

We have chosen instead to decompose the problem vertically as our primary way of slicing up the problem. Rather than slice the problem on the basis of internal workings of the solution we slice the problem on the basis of desired external manifestations of the robot control system.

To this end we have defined a number of *levels of competence* for an autonomous mobile robot. A level of competence is an informal specification of a desired class of behaviors for a robot over all environments it will encounter. A higher level of competence implies a more specific desired class of behaviors.

We have used the following levels of competence (an earlier version of these was reported in Brooks (1984*a*)) as a guide in our work:

0. Avoid contact with objects (whether the objects move or are stationary).

1. Wander aimlessly around without hitting things.
2. "Explore" the world by seeing places in the distance which look reachable and heading for them.
3. Build a map of the environment and plan routes from one place to another.
4. Notice changes in the "static" environment.
5. Reason about the world in terms of identifiable objects and perform tasks related to certain objects.
6. Formulate and execute plans which involve changing the state of the world in some desirable way.
7. Reason about the behavior of objects in the world and modify plans accordingly.

Notice that each level of competence includes as a subset each earlier level of competence. Since a level of competence defines a class of valid behaviors it can be seen that higher levels of competence provide additional constraints on that class.

2.2 Layers of Control

The key idea of levels of competence is that we can build layers of a control system corresponding to each level of competence and simply add a new layer to an existing set to move to the next higher level of overall competence.

We start by building a complete robot control system which achieves level *0* competence. It is debugged thoroughly. We never alter that system. We call it the zeroth level control system, Next we build a another control layer, which we call the first level control system. It is able to examine data from the level *0* system and is also permitted to inject data into the internal interfaces of level *0* suppressing the normal data flow. This layer, with the aid of the zeroth, achieves level *1* competence. The zeroth layer continues to run unaware of the layer above it which sometimes interferes with its data paths.

The same process is repeated to achieve higher levels of competence. See Figure 3.

We call this architecture a *subsumption architecture*.

In such a scheme we have a working control system for the robot very early in the piece—as soon as we have built the first layer. Additional layers can be added later, and the initial working system need never be changed.

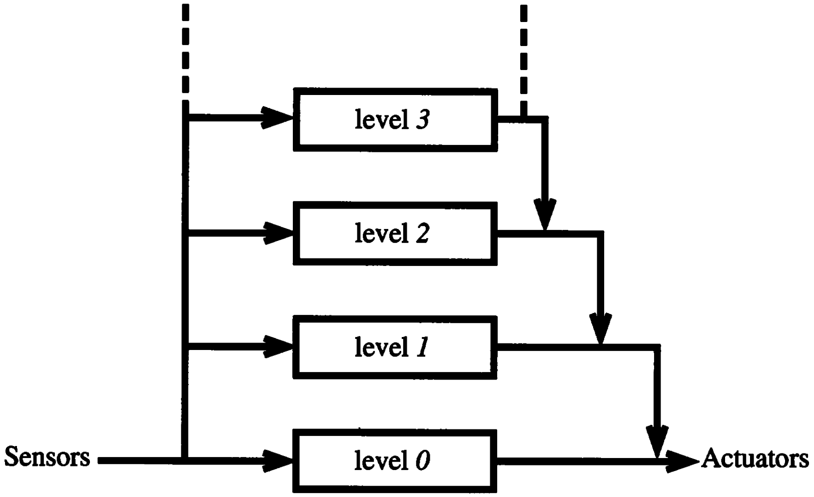

Figure 3: Control is layered with higher level layers subsuming the roles of lower level layers when they wish to take control. The system can be partitioned at any level, and the layers below form a complete operational control system

We claim that this architecture naturally lends itself to solving the problems for mobile robots delineated in section 1.1.

Multiple Goals: Individual layers can be working on individual goals concurrently. The suppression mechanism then mediates the actions that are taken. The advantage here is that there is no need to make an early decision on which goal should be pursued. The results of pursuing all of them to some level of conclusion can be used for the ultimate decision.

Multiple Sensors: In part we can ignore the sensor fusion problem as stated earlier using a subsumption architecture. Not all sensors need to feed into a central representation. Indeed certain readings of all sensors need not feed into central representations—only those which perception processing identifies as extremely reliable might be eligible to enter such a central representation. At the same time however the sensor values may still be being used by the robot. Other layers may be processing them in some fashion and using the results to achieve their own goals, independent of how other layers may be scrutinizing them.

Robustness: Multiple sensors clearly add to the robustness of a system when their results can be used intelligently. There is another source of robustness in a subsumption architecture. Lower levels which have been well debugged continue to run when higher levels are added. Since a higher level can only suppress the outputs of lower levels by actively interfering with replacement data, in the cases that it can not produce results in a timely fashion the lower levels will still produce results which are sensible, albeit at a lower level of competence.

Extensibility: An obvious way to handle extensibility is to make each new layer run on its own processor. We will see below that this is practical as there are in general fairly low bandwidth requirements on communication channels between layers. In addition we will see that the individual layers can easily be spread over many loosely coupled processors.

2.3 The Structure of Layers

But what about building each individual layer? Don't we need to decompose a single layer in the traditional manner? This is true to some extent, but the key difference is that we don't need to account for all desired perceptions and processing and generated behaviors in a single decomposition. We are free to use different decompositions for different sensor-set task-set pairs.

We have chosen to build layers from a set of small processors which send messages to each other.

Each processor is a finite state machine with the ability to hold some data structures. Processors send messages over connecting "wires". There is no handshaking or acknowledgement of messages. The processors run completely asynchronously, monitoring their input wires, and sending messages on their output wires. It is possible for messages to get lost—it actually happens quite often. There is no other form of communication between processors, in particular there is no shared global memory.

All processors (which we refer to as modules) are created equal in the sense that within a layer there is no central control. Each module merely does its thing as best it can.

Inputs to modules can be suppressed and outputs can be inhibited by wires terminating from other modules. This is the mechanism by which higher level layers subsume the role of lower levels.

3 A Robot Control System Specification Language

There are two aspects to the components of our layered control architecture. One is the internal structure of the modules, and the second is the way in which they communicate. In this section we flesh out the details of the semantics of our modules and explain a description language for them.

3.1 Finite State Machines

Each module, or processor, is a finite state machine, augmented with some instance variables which can actually hold LISP data structures.

Each module has a number of input lines and a number of output lines. Input lines have single element buffers. The most recently arrived message is always available for inspection. Messages can be lost if a new one arrives on an input line before the last was inspected.

There is a distinguished input to each module called **reset**.

Each state is named. When the system first starts up all modules start in the distinguished state named **NIL**. When a signal is received on the reset line the module switches to state **NIL**. A state can be specified as one of four types.

Output	An output message, computed as a function of the module's input buffers and instance variables, is sent to an output line. A new specified state is then entered.
Side effect	One of the module's instance variables is set to a new value computed as a function of its input buffers and variables. A new specified state is then entered.
Conditional dispatch	A predicate on the module's instance variables and input buffers is computed and depending on the outcome one of two subsequent states is entered.
Event dispatch	An sequence of pairs of conditions and states to branch to are monitored until one of the events is true. The events are in combinations of arrivals of messages on input lines and the expiration of time delays.[1]

An example of a module defined in our specification language is the **avoid** module:

---

[1] The exact semantics are as follows. After an event dispatch is executed all input lines are monitored for message arrivals. When the next event dispatch is executed it has access to latches which indicate whether new messages arrived on each input line. Each condition is evaluated in turn. If it is true then the dispatch to the new state happens. Each condition is an and/or expression on the input line latches. In addition, condition expressions can include delay terms which become true a specified amount of time after the beginning of the execution of the event dispatch. An event dispatch waits until one of its condition expressions is true.

```
(defmodule avoid 1
  :inputs (force heading)
  :outputs (command)
  :instance-vars (resultforce)
  :states
    ((nil (event-dispatch (and force heading)
                          plan))
     (plan (setf resultforce
                 (select-direction force
                                   heading))
           go)
     (go (conditional-dispatch
             (significant-force-p resultforce
                                  1.0)
           start
           nil))
     (start (output command
                    (follow-force resultforce))
            nil)))
```

Here, each of `select-direction`, `significant-force-p` and `follow-force` are LISP functions, while `setf` is the modern LISP assignment special form.

The force input line inputs a force with magnitude and direction found by treating each point found by the sonars as the site of a repulsive force decaying as the square of distance. Function `select-direction` takes this and combines it with the input on the heading line considered as a motive force. It selects the instantaneous direction of travel by summing the forces acting on the robot. (This simple technique computes the tangent to the minimum energy path computed by Khatib (1983).)

Function `significant-force-p` checks whether the resulting force is above some threshold—in this case it determines whether the resulting motion would take less than a second. The dispatch logic then ignores such motions.

Function `follow-force` converts the desired direction and force magnitude into motor velocity commands.

This particular module is part of the level *1* (as indicated by the argument "1" following the name of the module) control system described below. It essentially does local navigation, making sure obstacles are avoided by diverting a desired heading away from obstacles. It does not deliver the robot to a desired location—that is the task of level *2* competence.

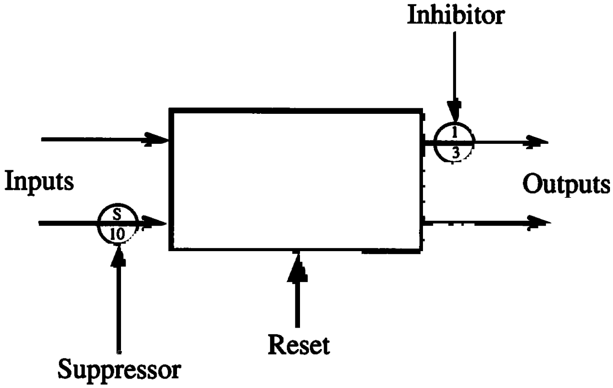

Figure 4: A module has input and output lines. Input signals can be suppressed and replaced with the suppressing signal. Output signals can be inhibited. A module can also be reset to state NIL.

3.2 Communication

Figure 4 shows the best way to think about these finite state modules for the purposes of communications. They have some input lines and some output lines. An output line from one module is connected to input lines of one or more other modules. One can think of these lines as wires, each with sources and a destination.

Additionally outputs may be inhibited, and inputs may be suppressed.

An extra wire can terminate (i.e., have its destination) at an output site of a module. If *any* signal travels along this wire it *inhibits* any output message from the module along that line for some pre-determined time. Any messages sent by the module to that output during that time period is lost.

Similarly an extra wire can terminate at an input site of a module. Its action is very similar to that of inhibition, but additionally, the signal on this wire, besides inhibiting signals along the usual path, actually gets fed through as the input to the module. Thus it *suppresses* the usual input and provides a replacement. If more than one suppressing wire is present they are essentially 'or'-ed together.

For both suppression and inhibition we write the time constants inside the circle.

In our specification language we write wires as a source (i.e., an output line) followed by a number of destinations (i.e., input lines). For instance the connection to the force input of the **avoid** module defined above might be the wire defined as:

```
(defwire 1 (feelforce force) (avoid force))
```

This links the force output of the **feelforce** module to the input of the **avoid** module in the level one control system.

Suppression and inhibition can also be described with a small extension to the syntax above. Below we see the suppression of the command input of the **turn** module, a level *0* module by a signal from the level *1* module **avoid**.

```
(defwire 1 (avoid command)
           ((suppress (turn command) 20.0)))
```

In a similar manner a signal can be connected to the reset input of a module.

4 A Robot Control System Instance

We have implemented a mobile robot control system to achieve levels *0* and *1* competence as defined above, and have started implementation of level *2* bringing it to a stage which exercises the fundamental subsumption idea effectively. We need more work on an early vision algorithm to complete level *2*.

4.1 Zeroth Level

The lowest level layer of control makes sure that the robot does not come into contact with other objects. It thus achieves level *0* competence. See Figure 5. If something approaches the robot it will move away. If in the course of moving itself it is about to collide with an object it will halt. Together these two tactics are sufficient for the robot to flee from moving obstacles, perhaps requiring many motions, without colliding with stationary obstacles. The combination of the tactics allows the robot to operate with with very coarsely calibrated sonars and a wide range of repulsive force functions. Theoretically, the robot is not invincible of course, and a sufficiently fast moving object, or a very cluttered environment might result in a collision. Over the course of a number of hours of autonomous operation, our physical robot (see section 5.2) has not collided with either a moving or fixed obstacle. The moving obstacles have, however, been careful to move slowly.

- The **turn** and **forward** modules communicate with the actual robot. They have extra communication mechanisms, allowing

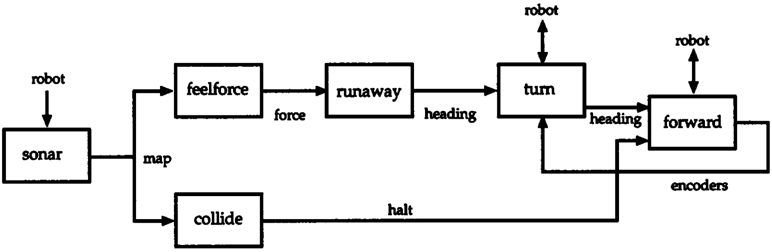

Figure 5: The level *0* control system.

them to send and receive commands to and from the physical robot directly. The **turn** module receives a heading specifying an inplace turn angle followed by a forward motion of a specified magnitude. It commands the robot to turn (and at the same time sends a busy message on an additional output channel illustrate in Figure 7) and on completion passes on the heading to the **forward** module (and also reports the shaft encoder readings on another output line shown in Figure 7) then goes into a wait state ignoring all incoming messages. The **forward** module commands the robot to move forward, but halts it if it receives and message on its halt input line during the motion. As soon as the robot is idle its sends out the shaft encoder readings–the message acts as a reset for the **turn** module, which is then once again ready to accept a new motion command. Notice the any heading commands sent to the **turn** module during transit are lost.

- The **sonar** module takes a vector of sonar readings, filters them for invalid readings, and effectively produces a robot centered map of obstacles in polar coordinates.

- The **collide** module monitors the sonar map and if it detects objects dead ahead it sends a signal on the halt line to the **motor** module. The **collide** module does not know or care whether the robot is moving. Halt messages sent while the robot is stationary are essentially lost.

- The **feelforce** module sums the results of considering each detected object as a repulsive force, generating a single resultant force.

- The **runaway** module monitors the 'force' produced by the sonar detected obstacles and sends commands to the **turn** module if it ever becomes significant.

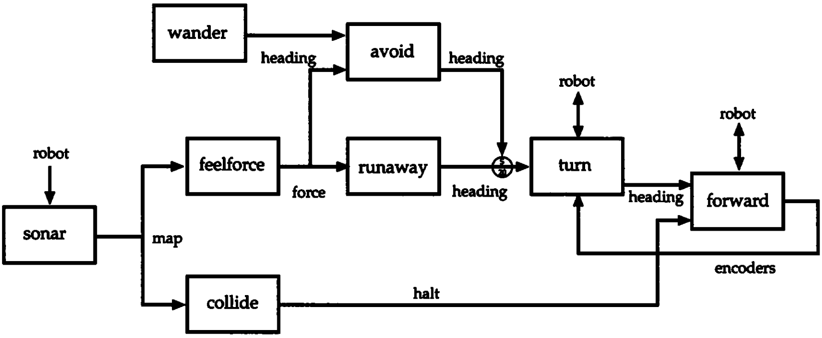

Figure 6: The level *0* control system augmented with the level *1* system.

Figure 5 gives a complete description of how the modules are connected together.

4.2 First Level

The first level layer of control, when combined with the zeroth, imbues the robot with the ability to wander around aimlessly without hitting obstacles. This was defined earlier as level *1* competence. This control level relies in a large degree on the zeroth level's aversion to hitting obstacles. In addition it uses a simple heuristic to plan ahead a little in order to avoid potential collisions which would need to be handled by the zeroth level.

- The **wander** module generates a new heading for the robot every 10 seconds or so.

- The **avoid** module, described in more detail in section 3, takes the result of the force computation from the zeroth level, and combines it with the desired heading to produce a modified heading which usually points in roughly the right direction, but is perturbed to avoid any obvious obstacles. This computation implicitly subsumes the computations of the **runaway** module, in the case that there is also a heading to consider. In fact the output of the **avoid** module suppresses the output from the **runaway** module as it enters the **motor** module.

Figure 6 gives a complete description of how the modules are connected together. Note that it is simply Figure 5 with some more modules and wires added.

4.3 Second Level

Level two is meant to add an exploratory mode of behavior to the robot, using visual observations to select interesting places to visit. A vision module finds corridors of free space. Additional modules provide a means of position servoing the robot to along the corridor despite the presence of local obstacles on its path (as detected with the sonar sensing system). The wiring diagram is shown in Figure 7. Note that it is simply Figure 6 with some more modules and wires added.

- The **status** module monitors the **turn** and **forward** modules. It maintains one status output which sends either *hi* or *lo* messages to indicate whether the robot is busy. In addition, at the completion of every turn and roll forward combination it sends out a combined set of shaft encoder readings.

- The **whenlook** module monitors the busy line from the **status** module, and whenever the robot has been sitting idle for a few seconds it decides its time to look for a corridor to traverse. It inhibits wandering so it can take some pictures and process them without wandering away from its current location, and resets the **pathplan** and **integrate** modules–this latter action ensures that it will know how far it has moved from its observation point should any **runaway** impulses perturb it.

- The **look** module initiates the vision processing, and waits for a candidate freeway. It filters out poor candidates and passes any acceptable one to the **pathplan** module.

- The **stereo** module is supposed to use stereo TV images (Grimson 1985), obtained by the robot, to find a corridor of free space. At the time of writing final version of this module had not been implemented. Instead, both in simulation and on the physical robot, we have replaced it with a sonar-base corridor finder.

- The **integrate** module accumulates reports of motions from the **status** module and always sends its most recent result out on its integral line. It gets restarted by application of a signal to its reset input.

- The **pathplan** module takes a goal specification (in terms of an angle to turn, a distance to travel) and attempts to reach that goal. To do this it sends headings to the **avoid** module, which may perturb them to avoid local obstacles, and monitors its integral input which is an integration of actual motions. The messages to

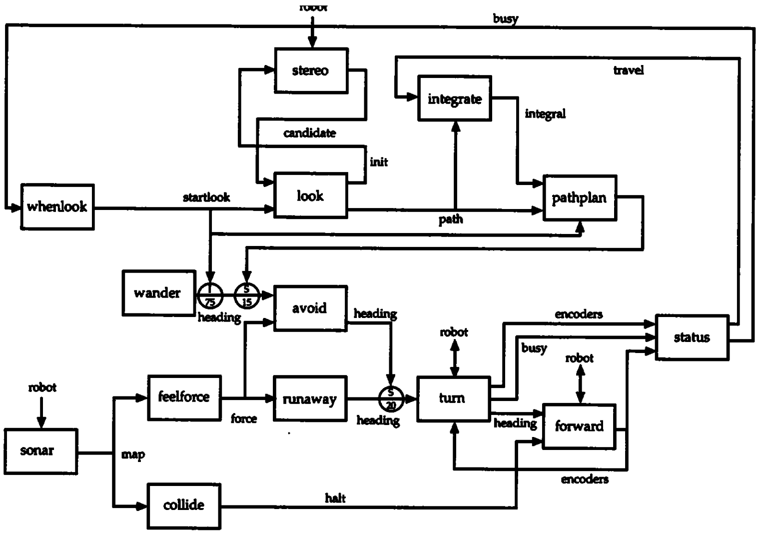

Figure 7: The level *0* and *1* control systems augmented with the level *2* system.

the **avoid** module suppress random wanderings of the robot, so long as the higher level planner remains active. When the position of the robot is close to the desired position (the robot is unaware of control errors due to wheel slippage etc., so this is a dead reckoning decision) it terminates.

The current wiring of the second level of control is shown in Figure 7, augmenting the two lower level control systems. The zeroth and first layers still play an active role during normal operation of the second layer.

5 Performance

The control system described here has been used extensively to control both a simulated robot and an actual physical robot wandering around a cluttered laboratory and a machine room.

5.1 A Simulated Robot

The simulation tries to simulate all the errors and uncertainties that exist in the world of the real robot. When commanded to turn through angle α and travel distance d the simulated robot actually turns through

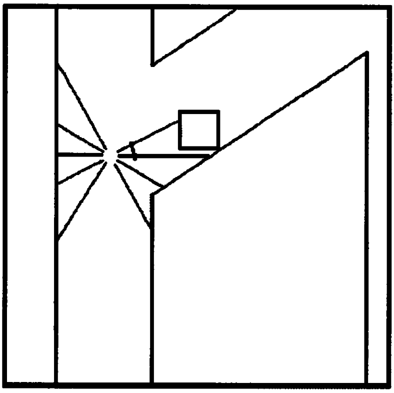

Figure 8: The simulated robot receives 12 sonar readings. Some sonar beams glance off walls and do not return within a certain time.

angle $\alpha + \delta\alpha$ and travels distance $d + \delta d$. Its sonars can bounce off walls multiple times, and even when they do return they have a noise component in the readings modeling thermal and humidity effects. We feel it is important to have such a realistic simulation. Anything less leads to incorrect control algorithms.

The simulator runs off a clock and runs at the same rate as would the actual robot. It actually runs on the same processor that is simulating the subsumption architecture. Together they are nevertheless able to perform a realtime simulation of the robot and its control and also drive graphics displays of robot state and module performance monitors. Figure 8 shows the robot (which itself is not drawn) receiving sonar reflections at some of its 12 sensors. Other beams did not return within the time allocated for data collection. The beams are being reflected by various walls. There is a small bar in front of the robot perpendicular to the direction the robot is pointing.

Figure 9 shows an example world in two dimensional projection. The simulated robot with a first level control system connected was allowed to wander from an initial position. The squiggly line traces out its path. Note that it was wandering aimlessly and that it hit no obstacles.

Figure 10 shows two examples of the same scene and the motion of the robot with the second level control system connected. In these cases the **stereo** module was supplanted with a situation specific module which gave out two precise corridor descriptions. While achieving the goals of following these corridors the lower level wandering behavior was suppressed. However the obstacle avoiding behavior of the lower

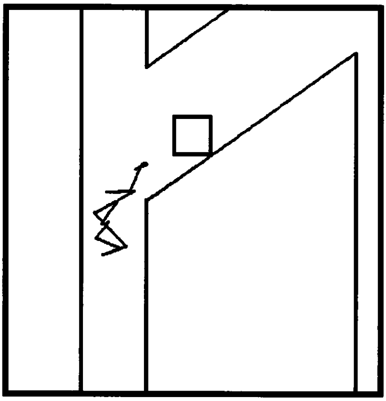

Figure 9: Under levels *0* and *1* control the robot wanders around aimlessly. It does not hit obstacles.

levels continued to function–in both cases the robot avoided the square obstacle. The goals were not reached exactly. The simulator models a uniformly distributed error of ±5% in both turn and forward motion. As soon as the goals had been achieved satisfactorily the robot reverted to its wandering behavior.

5.2 A Physical Robot

We have constructed a mobile robot shown in Figure 11. It is about 17 inches in diameter and about 30 inches from the ground to the top platform. Most of the processing occurs offboard on a LISP MACHINE.

The drive mechanism was purchased from Real World Interface of Sudbury, Massachusetts. Three parallel drive wheels are steered together. The two motors are servoed by a single microprocessor. The robot body is attached to the steering mechanism and always points in the same direction as the wheels. It can turn in place (actually it inscribes a circle about 1 cm in diameter).

Currently installed sensors are a ring of twelve Polaroid sonar time of flight range sensors and two Sony CCD cameras. The sonars are arranged symmetrically around the rotating body of the robot. The cameras are on a tilt head (pan is provided by the steering motors). We plan to install feelers which can sense objects at ground level about six inches from the base extremities.

A central cardcage contains the main onboard processor, an Intel 8031. It communicates with offboard processors via a 12Kbit/sec du-

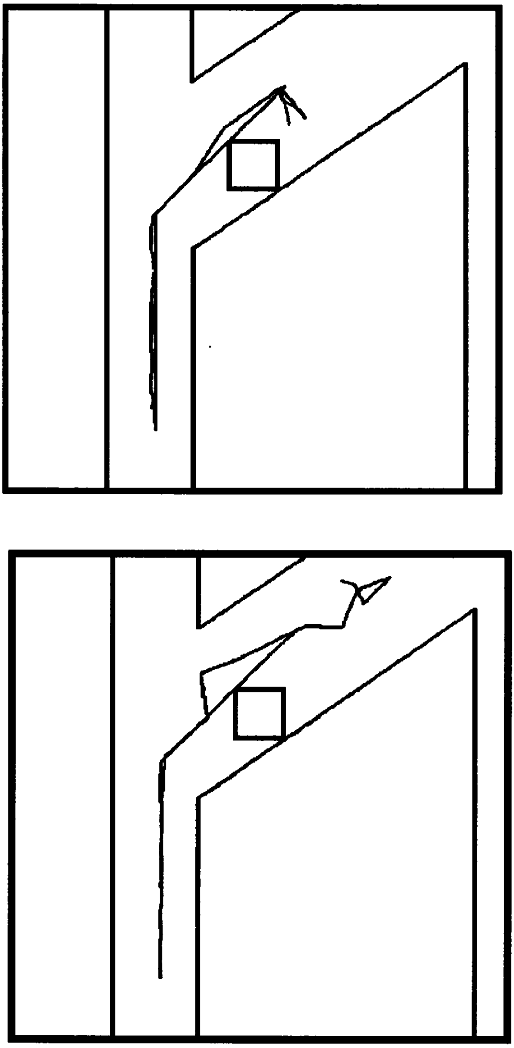

Figure 10: With level *2* control the robot tries to achieve commanded goals. The nominal goals are the two straight lines. After reaching the second goal, since there are no new goals forthcoming, the robot reverts to aimless level *1* behavior.

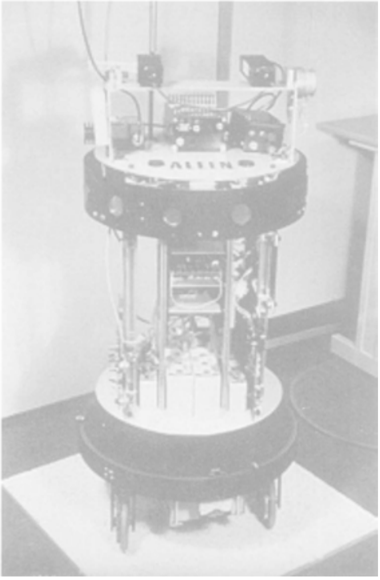

Figure 11: The MIT AI Lab mobile robot.

plex radio link. The radios are modified Motorola digital voice encryption units. Error correction cuts the effective bit rate to less than half the nominal rating. The 8031 passes commands down to the motor controller processor and returns encoder readings. It controls the sonars, and the tilt head and switches the cameras through a single channel video transmitter mounted on top of the robot. The latter transmits a standard TV signal to a LISP MACHINE equipped with a demodulator and frame grabber.

The robot has spent a few hours wandering around a laboratory and a machine room.

Under level *0* control the robot finds a large empty space and then sits there contented until a moving obstacle approaches. Two people together can successfully herd the robot just about anywhere–through doors or between rows of disk drives, for instance.

When level *1* control is added the robot is no longer content to sit in an open space. After a few seconds it heads off in a random direction. Our uncalibrated sonars and obstacle repulsion functions make it overshoot a little to locations where the **runaway** module reacts. It would be interesting to make this the basis of adaption of certain parameters.

Under level *2* a sonar-based corridor finder usually finds the most distant point in the room. The robot heads of in the direction. People walking in front of the robot cause it to detour, but still get to the initially desired goal even when it involves squeezing between closely spaced obstacles. If the sonars are in error and a goal is selected beyond a wall, say, the robot usually ends up in a position where the attractive force of the goal is within a threshold used by **avoid** of the repulsive forces of the wall. At this point **avoid** does not issue any heading, as it would be for some trivial motion of the robot. The robot sits still defeated by the obstacle. The **whenlook** module, however, notices that the robot is idle and initiates a new scan for another corridor of free space to follow.

5.3 Implementation Issues

While we have been able to simulate sufficient processors on a single LISP MACHINE up until now, that capability will soon pass as we bring on line our vision work (the algorithms have been debugged as traditional serial algorithms but we plan on re-implementing them within the subsumption architecture). Building the architecture in custom chips is a long term goal.

One of the motivations for developing the layered control system was extensibility of processing power. The fact that it is decomposed into asynchronous processors with low bandwidth communication and no shared memory should certainly assist in achieving that goal. New processors can simply be added to the network by connecting their inputs and outputs at appropriate places—there are no bandwidth or synchronization considerations in such connections.

The finite state processors need not be large. Sixteen states is more than sufficient for all modules we have so far written. (Actually 8 states are sufficient under the model of the processors we have presented here and used in our simulations. However we have refined the design somewhat towards gate level implementation and there we use simpler more numerous states.) Many such processors could easily be packed on a single chip.

The LISP programs that are called by the finite state machines are all rather simple. We believe it is possible to implement each of them with a simple network of comparators, selectors, polar coordinate vector adders and monotonic function generators. The silicon area overhead for each module would probably not be larger than that required for the finite state machine itself.

6 Conclusion

The key ideas in this paper are:

- The mobile robot control problem can be decomposed in terms of behaviors rather than in terms of functional modules.
- It provides a way to incrementally build and test a complex mobile robot control system.
- Useful parallel computation can be performed on a low bandwidth loosely coupled network of asynchronous simple processors. The topology of that network is relatively fixed.
- There is no need for a central control module of a mobile robot. The control system can be viewed as a system of agents each busy with their own solipsist world.

Besides leading to a different implementation strategy it is also interesting to note the way the decomposition affected the capabilities of the robot control system we have built. In particular our control system deals with moving objects in the environment at the very lowest level, and has a specific module (**runaway**) for that purpose. Traditionally mobile robot projects have delayed handling moving objects in the environment beyond the scientific life of the project.
Note: A drawback of the presentation in this paper was a merging of the algorithms for control of the robot from the implementation medium. We felt this was necessary to convince the reader of the utility of both. It is unlikely that the subsumption architecture would appear to be useful without a clear demonstration of how a respectable and useful algorithm can run on it. Mixing the two descriptions as we have done demonstrates the proposition.

7 Acknowledgements

Tomás Lozano-Pérez, Eric Grimson, Jon Connell and Anita Flynn have all provided helpful comments on earlier drafts of this paper.

This work was supported in part by an IBM Faculty Development Award, in part by a grant from the Systems Development Foundation, in part by an equipment grant from Motorola, and in part by the Advanced Research Projects Agency under Office of Naval Research contracts N00014–80–C–0505 and N00014–82–K–0334.

Bibliography

Agre, P. E. (1988), The Dynamic Structure of Everyday Life, Technical Report 1085, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Agre, P. E. (1991), *The Dynamic Structure of Everyday Life*, Cambridge University Press, Cambridge, United Kingdom.

Agre, P. E. & Chapman, D. (1987), "Pengi: An Implementation of a Theory of Activity", in *Proceedings of the Sixth Annual Meeting of the American Association for Artificial Intelligence*, Morgan Kaufmann Publishers, Seattle, Washington, pp. 268–272. Also appeared in Luger (1995).

Agre, P. E. & Chapman, D. (1990), "What Are Plans For?", *Robotics and Autonomous Systems* **6**(1,2), 17–34. Also appeared in Maes (1990*b*).

Agre, P. E. & Chapman, D. (n.d.), "Unpublished memo", MIT Artificial Intelligence Lab. Cambridge, Massachusetts.

Angle, C. M. (1989*a*), "Atilla's Crippled Brother Marvin", MIT Term Paper in Electrical Engineering and Computer Science.

Angle, C. M. (1989*b*), Genghis, a Six Legged Walking Robot, Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts. Bachelor's thesis.

Angle, C. M. & Brooks, R. A. (1990), "Small Planetary Rovers", in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Ikabara, Japan, pp. 383–388.

Arbib, M. A. (1964), *Brains, Machines and Mathematics*, McGraw-Hill, New York, New York.

Arkin, R. C. (1989), "Towards the Unification of Navigational Planning and Reactive Control", in *AAAI Spring Symposium on Robot Navigation and Working Notes*, pp. 1–5.

Arkin, R. C. (1990), "Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation", *Robotics and Autonomous Systems* **6**(1,2), 105–122. Also appeared in Maes (1990*b*).

Ashby, W. R. (1952), *Design for a Brain*, Chapman and Hall, London, United Kingdom.

Ashby, W. R. (1956), *An Introduction to Cybernetics*, Chapman and Hall, London, United Kingdom.

Atkeson, C. G. (1989), Using Local Models to Control Movement, in D. S. Touretzky, ed., *Neural Information Processing 2*, Morgan Kaufmann Publishers, Los Altos, California, pp. 316–324.

Ballard, D. H. (1989), "Reference Frames for Animate Vision", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, Detroit, Michigan, pp. 1635–1641.

Barrow, H. & Salter, S. (1970), Design of Low-Cost Equipment for Cognitive Robot Research, in B. Meltzer & D. Michie, eds, *Machine Intelligence 5*, American Elsevier Publishing, New York, New York, pp. 555–566.

Bassler, U. (1983), *Neural Basis of Elementary Behavior in Stick Insects*, Springer-Verlag.

Beer, R. D. (1990), *Intelligence as Adaptive Behavior*, Academic Press, San Diego, California.

Bergland, R. (1985), *The Fabric of Mind*, Viking, New York, New York.

Bizzi, E. (1980), Central and peripheral mechanisms in motor control, in G. Stelmach & J. Requin, eds, *Tutorials in Motor Behavior*, North-Holland.

Bloom, F. (1976), "Endorphins: Profound Behavioral Effects", *Science* **194**, 630–634.

Bobrow, R. J. & Brown, J. S. (1975), Systematic Understanding: Synthesis, Analysis, and Contingent Knowledge in Specialized Understanding Systems, in Bobrow & Collins, eds, *Representation and Understanding*, Academic Press, New York, New York, pp. 103–129.

Brachman, R. J. & Levesque, H. J. (1985), *Readings in Knowledge Representation*, Morgan Kaufmann Publishers, Los Altos, California.

Brady, D. (1990), "Switching Arrays Make Light Work in a Simple Processor", *Nature* **344**, 486–487.

Brady, M., Hollerbach, J., Johnson, T. & Lozano-Pérez, T., eds (1982), *Robot Motion: Planning and Control*, MIT Press, Cambridge, Massachusetts.

Braitenberg, V. (1984), *Vehicles: Experiments in Synthetic Psychology*, MIT Press, Cambridge, Massachusetts.

Brooks, R. A. (1981), "Symbolic Reasoning Among 3-D Models and 2-D Images", *Artificial Intelligence Journal* **17**, 285–348.

Brooks, R. A. (1982), "Symbolic Error Analysis and Robot Planning", *International Journal of Robotics Research* **1**(4), 29–68.

Brooks, R. A. (1984*a*), Aspects of mobile robot visual map making, in Hanafusa & Inoue, eds, *Robotics Research 2*, MIT, Cambridge, Massachusetts, pp. 369–375.

Brooks, R. A. (1984*b*), *Model-Based Computer Vision*, UMI Research Press, Ann Arbor, Michigan.

Brooks, R. A. (1985), "Visual Map Making for a Mobile Robot", in *Proceedings of the 1985 IEEE Conference on Robotics and Automation*, pp. 824–829.

Brooks, R. A. (1986), "A Robust Layered Control System for a Mobile Robot", *IEEE Journal of Robotics and Automation* **RA-2**, 14–23.

Brooks, R. A. (1987), "Micro-Brains for Micro-Brawn: Autonomous Microrobots", in *IEEE Micro Robots and Teleoperators Workshop*, Hyannis, Massachusetts.

Brooks, R. A. (1989), "A Robot That Walks: Emergent Behavior from a Carefully Evolved Network", *Neural Computation* **1**(2), 253–262.

Brooks, R. A. (1990*a*), The Behavior Language User's Guide, Memo 1227, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Brooks, R. A. (1990*b*), "Challenges for Complete Creature Architectures", in *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, Massachusetts, pp. 434–443.

Brooks, R. A. (1990*c*), "Elephants Don't Play Chess", *Robotics and Autonomous Systems* **6**(1,2), 3–15. Also appeared in Maes (1990*b*).

Brooks, R. A. (1991*a*), "Integrated Systems Based on Behaviors", *SIGART, Special Issue on Integrated Intelligent Systems.*

Brooks, R. A. (1991*b*), "Intelligence Without Reason", in *Proceedings of the 1991 International Joint Conference on Artificial Intelligence*, pp. 569–595. Also appeared in Steels & Brooks (1995).

Brooks, R. A. (1991*c*), "Intelligence Without Representation", *Artificial Intelligence Journal* **47**, 139–160. Also appeared in Luger (1995).

Brooks, R. A. & Connell, J. (1986), "Asynchronous Distributed Control System for a Mobile Robot", in *SPIE's Cambridge Symposium on Optical and Opto-Electronic Engineering Proceedings, Vol. 727*, pp. 77–84.

Brooks, R. A. & Flynn, A. M. (1989), "Robot Beings", in *IEEE/RSJ International Workshop on Intelligent Robots and Systems*, Tsukuba, Japan, pp. 2–10.

Brooks, R. A., Connell, J. H. & Ning, P. (1988), Herbert: A Second Generation Mobile Robot, Memo 1016, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Brooks, R. A., Flynn, A. M. & Marill, T. (1987), Self Calibration of Motion and Stereo for Mobile Robot Navigation, Memo 984, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Campbell, J. (1983), Go, in M. Bramer, ed., *Computer Game-Playing: Theory and Practice*, Ellis Horwood, Chichester, United Kingdom.

Chapman, D. (1987), "Planning for Conjunctive Goals", *Artificial Intelligence Journal* **32**, 333–377.

Chapman, D. (1990), Vision, Instruction and Action, Technical Report 1085, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Charniak, E. & McDermott, D. (1984), *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, Massachusettse.

Chatila, R. & Laumond, J. (1985), "Position Referencing and Consistent World Modeling for Mobile Robots", in *IEEE International Conference on Robotics and Automation*, pp. 138–145.

Churchland, P. S. (1986), *Neurophilosophy: Toward a Unified Science of the Mind/Brain*, MIT Press, Cambridge, Massachusetts.

Cohen, L. & Wu, J.-Y. (1990), "One Neuron, Many Units?", *Nature* **346**, 108–109.

Cohen, P. R. (1991), "A Survey of the Eigth National Conference on Artificial Intelligence: Pulling Together or Pulling APart?", *AI Magazine*.

Condon, J. & Thompson, K. (1984), Belle, in P. Frey, ed., *Chess Skill in Man and Machine*, Springer-Verlag.

Connell, J. H. (1987), "Creature Building with the Subsumption Architecture", in *Proceedings of the International Joint Conference on Artificial Intelligence*, Milan, Italy, pp. 1124–1126.

Connell, J. H. (1988), A Behavior-Based Arm Controller, Memo 1025, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Connell, J. H. (1989), A Colony Architecture for an Artificial Creature, Technical Report 1151, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Cox, I. & Wilfong, G., eds (1990), *Autonomous Robot Vehicles*, Springer Verlag, New York, New York.

Crowley, J. I. (1985), "Navigation for an intelligent mobile robot", *IEEE Journal of Robotics and Automation* **RA-1**(1), 31–41.

Cruse, H. (1990), "What Mechanisms Coordinate Leg Movement in Walking Arthropods?", *Trends in Neurosciences* **13**(1), 15–21.

de Kleer, J. & Brown, J. S. (1984), "A Qualitative Physics Based on Confluences", *Artificial Intelligence Journal* **24**, 7–83.

Dennett, D. C. & Kinsbourne, M. (1990), Time and the Observer: the Where and When of Consciousness in the Brain, Technical report, Center for Cognitive Studies, Tufts University.

Dickmanns, E. D. & Graefe, V. (1988), "Dynamic Monocular Machine Vision", *Machine Vision and Applications* **1**, 223–240.

Dreyfus, H. L. (1981), From Micro-Worlds to Knowledge Representation: AI at an Impasse, in J. Haugeland, ed., *Mind Design*, MIT Press, Cambridge, Massachusetts, pp. 161–204.

Drumheller, M. (1985), Robot Localization Using Range Data, Master's thesis, MIT Dept. of Mechanical Engineering, Cambridge, Massachusetts. Bachelor's thesis.

Eichenbaum, H., Wiener, S., Shapiro, M. & Cohen, N. (1989), "The organization of spatial coding in the hippocampus: A study of neural ensemble activity", *Journal of Neuroscience* **9**(8), 2764–2775.

Elfes, A. & Talukdar, S. (1983), "A distributed control system for the CMU rover", in *Proceedings IJCAI-83*, pp. 830–833.

Ernst, H. A. (1961), A Computer-Operated Mechanical Hand, PhD thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Evans, T. G. (1968), A Program for the Solution of Geometric-Analogy Intelligence Test Questions, in M. Minsky, ed., *Semantic Information Processing*, MIT Press, Cambridge, Massachusetts, pp. 271–353.

Fahlman, S. E. (1974), "A Planning System for Robot Construction", *Artificial Intelligence Journal* **5**, 1–50.

Feigenbaum, E. A. & Feldman, J. (1963), *Computers and Thought*, McGraw-Hill, New York, New York.

Firby, R. J. (1989), Adaptive Execution in Dynamic Domains, PhD thesis, Yale University.

Flynn, A. M. (1985), Redundant sensors for mobile robot navigation, Master's thesis, MIT, Cambridge, Massachusetts.

Flynn, A. M. (1988), "Gnat Robots: A Low-Intelligence, Low-Cost Approach", in *IEEE Solid-State Sensor and Actuator Workshop*, Hilton Head, South Carolina.

Flynn, A. M. (1989), "Gnat Robots (And How They Will Change Robotics)", in *IEEE Micro Robots and Teleoperators Workshop*, Hyannis, Massachusetts.

Flynn, A. M., Brooks, R. A. & Tavrow, L. S. (1989*a*), Twilight Zones and Cornerstones: A Gnat Robot Double Feature, Memo 1126, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Flynn, A. M., Brooks, R. A., Wells, W. M. & Barrett, D. S. (1989*b*), "The World's Largest One Cubic Inch Robot", in *Proceedings IEEE Micro Electro Mechanical Systems*, Salt Lake City, Utah, pp. 98–101.

Gazzaniga, M. S. & LeDoux, J. E. (1977), *The Integrated Mind*, Plenum, New York, New York.

Gibbs, H. (1985), *Optical Bistability: Controlling Light with Light*, Academic Press, New York, New York.

Giralt, G., Chatila, R. & Vaisset, M. (1984), An Integrated Navigation and Motion Control System for Autonomous Multisensory Robots, in Brady & Paul, eds, *Robotics Research 1*, MIT Press, Cambridge, Massachusetts, pp. 191–214. Also appeared in Cox & Wilfong (1990).

Götz, K. G. & Wenking, H. (1973), "Visual Control of Locomotion in the Walking Fruitfly *Drosophilia*", *Journal of Comparative Physiology* **85**, 235–266.

Gould, J. L. & Marler, P. (1986), "Learning by Instinct", *Scientific American* pp. 74–85.

Gould, S. & Eldredge, N. (1977), "Punctuated Equilibria: The Tempo and Mode of Evolution Reconsidered", *Paleobiology* **3**, 115–151.

Greenblatt, R., Eastlake, D. & Crocker, S. (1967), "The Greenblatt Chess Program", in *Am. Fed. Inf. Proc. Soc. Conference Proceedings*, pp. 801–810.

Grimson, W. E. L. (1985), "Computational experiments with a feature based stereo algorithm", *IEEE Trans. PAMI* **PAMI-7**, 17–34.

Grimson, W. E. L. (1990), *Object Recognition by Computer: The Role of Geometric Constraints*, MIT Press, Cambridge, Massachusetts.

Hartmanis, J. (1971), "Computational Complexity of Random Access Stored Program Machines", *Mathematical Systems Theory* **5**(3), 232–245.

Hayes, P. J. (1985), The Second Naive Physics Manifesto, in J. R. Hobbs & R. C. Moore, eds, *Formal Theories of the Commonsense World*, Ablex, Norwood, New Jersey, pp. 1–36. Also appeared in Luger (1995).

Hillis, W. D. (1985), *The Connection Machine*, MIT Press, Cambridge, Massachusetts.

Hodges, A. (1983), *Alan Turing: The Enigma*, Simon and Shuster, New York, New York.

Hofstadter, D. R. & Dennett, D. C. (1981), *The Mind's I*, Bantam Books, New York, New York.

Horswill, I. D. & Brooks, R. A. (1988), "Situated Vision in a Dynamic World: Chasing Objects", in *Proceedings of the Seventh Annual Meeting of the American Association for Artificial Intelligence*, St. Paul, Minnesota, pp. 796–800.

Hsu, F.-H., Anantharaman, T., Campbell, M. & Nowatzyk, A. (1990), "A Grandmaster Chess Machine", *Scientific American* **263**(4), 44–50.

Johnson, M. (1987), *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*, The University of Chicago Press, Chicago, Illinois.

Kaelbling, L. P. (1990), Learning in Embedded Systems, PhD thesis, Stanford University.

Kaelbling, L. P. & Rosenschein, S. J. (1990), "Action and Planning in Embedded Agents", *Robotics and Autonomous Systems* **6**(1,2), 35–48. Also appeared in Maes (1990*b*).

Kanayama, Y. (1983), "Concurrent programming of intelligent robots", in *Proceedings of IJCAI-83*, pp. 834–838.

Khatib, O. (1983), "Dynamic control of manipulators in operational space", in *Sixth IFTOMM Cong. Theory of Machines and Mechanisms*.

Klein, C., Olson, K. & Pugh, D. (1983), "Use of force and attitude sensors for locomotion of a legged vehicle", *International Journal of Robotics Research* **2**(2), 3–17.

Knuth, D. E. & Moore, R. E. (1975), "An Analysis of Alpha-Beta Pruning", *Artificial Intelligence Journal* **6**, 293–326.

Kravitz, E. A. (1988), "Hormonal Control of Behavior: Amines and the Biasing Behavioral Output in Lobsters", *Science* **241**, 1775–1781.

Kreithen, M. (1983), Orientational strategies in birds: a tribute to W.T. Keeton, in *Behavioral Energetics: The Cost of Survival in Vertebrates*, Ohio State University, Columbus, Ohio, pp. 3–28.

Kuhn, T. S. (1970), *The Structure of Scientific Revolutions*, second, enlarged edn, The University of Chicago Press, Chicago, Illinois.

Kuipers, B. (1987), "A Qualitative Approach To Robot Exploration and Map Learning", in *AAAI Workshop on Spatial Reasoning and Multi-Sensor Fusion*.

Lenat, D. B. & Feigenbaum, E. A. (1991), "On the Thresholds of Knowledge", *Artificial Intelligence Journal* **47**, 185–250.

Lewis, A. C. (1986), "Memory Constraints and Flower Choice in Pieris Rapae", *Science* **232**, 863–865.

Lozano-Pérez, T. & Brooks, R. A. (1984), An Approach to Automatic Robot Programming, in M. S. Pickett & J. W. Boyse, eds, *Solid Modeling by Computers*, Plenum Press, New York, New York, pp. 293–328.

Lozano-Pérez, T., Jones, J. L., Mazer, E. & O'Donnell, P. A. (1989), "Task-Level Planning of Pick-and-Place Robot Motions", *IEEE Computer* **22**(3), 21–29.

Lozano-Pérez, T., Mason, M. W. & Taylor, R. H. (1984), "Automatic Synthesis of Fine-Motion Strategies for Robots", *International Journal of Robotics Research* **3**(1), 3–24.

Luger, G. F., ed. (1995), *Computation & Intelligence: Collected Readings*, AAAI Press, MIT Press, Menlo Park, California.

Maes, P. (1989), "The Dynamics of Action Selection", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Morgan Kaufmann Publishers, Detroit, Michigan, pp. 991–997.

Maes, P. (1990*a*), "Situated Agents Can Have Goals", *Robotics and Autonomous Systems* **6**(1,2), 49–70. Also appeared in Maes (1990*b*) and in the special issue of the Journal of Robotics and Autonomous Systems, Spring '90, North Holland.

Maes, P. & Brooks, R. A. (1990), "Learning to Coordinate Behaviors", in *Proceedings of the Eighth Annual Meeting of the American Association for Artificial Intelligence*, MIT Press, Cambridge, Massachusetts, pp. 796–802.

Maes, P., ed. (1990*b*), *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back*, MIT Press, Cambridge, Massachusetts.

Mahadevan, S. & Connell, J. (1990), Automatic Programming of Behavior-Based Robots Using Reinforcement Learning, Technical report, IBM T.J. Watson Research Report.

Malcolm, C. & Smithers, T. (1990), "Symbol Grounding Via a Hybrid Architecture in an Autonomous Assembly System", *Robotics and Autonomous Systems* **6**(1,2), 123–144. Also appeared in Maes (1990*b*).

Malkin, P. K. & Addanki, S. (1990), "LOGnets: A Hybrid Graph Spatial Representation for Robot Navigation", in *Proceedings of the Eighth National Conference on Artificial Intelligence (AAAI-90)*, MIT Press, Cambridge, Massachusetts, pp. 1045–1050.

Marr, D. (1982), *Vision*, W.H. Freeman, San Francisco, California.

Mataric, M. J. (1989), "Qualitative Sonar Based Environment Learning for Mobile Robots", in *SPIE 1989 Mobile Robots IV Proceedings*.

Mataric, M. J. (1990*a*), A Distributed Model for Mobile Robot Environment-Learning and Navigation, Technical Report 1228, Massachusetts Institute of Technology Artificial Intelligence Lab, Cambridge, Massachusetts.

Mataric, M. J. (1990*b*), "Environment Learning Using a Distributed Representation", in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pp. 402–406.

Mataric, M. J. (1990*c*), "Navigation with a Rat Brain: A Neurobiologically-Inspired Model for Robot Spatial Representation", in *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, Massachusetts, pp. 169–175.

Mataric, M. J. (1991), "Behavioral Synergy Without Explicit Integration", *SIGART, Special Issue on Integrated Intelligent Systems*.

McCarthy, J. (1960), "Recursive Functions of Symbolic Expressions", *Communications of the Association for Computing Machinery* **3**, 184–195.

McCarthy, R. A. & Warrington, E. K. (1988), "Evidence for Modality-Specific Systems in the Brain", *Nature* **334**, 428–430.

McCorduck, P. (1979), *Machines Who Think*, Freeman, New York, New York.

McCulloch, W. & Pitts, W. (1943), "A Logical Calculus of the Ideas Immanent in Nervous Activity", *Bulletin of Mathematical Biophysics* **5**, 115–137.

McDermott, D. (1987). Position paper for DARPA Santa Cruz panel on interleaving planning and execution.

McFarland, D. (1985), *Animal Behaviour*, Benjamin/Cummings, Menlo Park, California.

McFarland, D. (1988), *Problems of Animal Behaviour*, Longman, Harlow, United Kingdom.

Michie, D. & Ross, R. (1970), Experiments with the Adaptive Graph Traverser, in B. Meltzer & D. Michie, eds, *Machine Intelligence 5*, American Elsevier Publishing, New York, New York, pp. 301–318.

Minsky, M. (1954), Neural Nets and the Brain Model Problem, PhD thesis, Princeton University, Univeristy Microfilms, Ann Arbor, Michigan. Unpublished Ph.D. dissertation.

Minsky, M. (1961), "Steps Toward Artificial Intelligence", *Proceedings IRE* **49**, 8–30. Also appeared in Feigenbaum & Feldman (1963) and Luger (1995).

Minsky, M. (1963), A Selected Descriptor-Indexed Bibliography to the Literature of Artificial Intelligence, in E. A. Feigenbaum & J. Feldman, eds, *Computers and Thought*, McGraw-Hill, New York, New York, pp. 453–523.

Minsky, M. (1986), *The Society of Mind*, Simon and Schuster, New York, New York.

Minsky, M. & Papert, S. (1969), *Perceptrons*, MIT Press, Cambridge, Massachusetts.

Minsky, M., ed. (1968), *Semantic Information Processing*, MIT Press, Cambridge, Massachusetts.

Mitchell, T. M. (1990), "Becoming Increasingly Reactive", in *Proceedings of the Eighth Annual Meeting of the American Association for Artificial Intelligence*, MIT Press, Cambridge, Massachusetts, pp. 1051–1058.

Moravec, H. P. (1981), *Robot Rover Visual Navigation*, UMI Research Press, Ann Arbor, Michigan.

Moravec, H. P. (1983), "The Stanford Cart and the CMU Rover", *Proceedings of the IEEE* **71**(7), 872–884. Also appeared in Cox & Wilfong (1990).

Moravec, H. P. (1984*a*), "Locomotion, Vision and Intelligence", in Brady & Paul, eds, *Proceedings of the First International Symposium on Robotics Research*, MIT Press, pp. 215–224.

Moravec, H. P. (1984*b*), Locomotion, Vision and Intelligence, in Brady & Paul, eds, *Robotics Research 1*, MIT Press, Cambridge, Massachusetts, pp. 215–224.

Moravec, H. P. (1987), "Sensor Fusion in Certainty Grids for Mobile Robots", in *IEEE Workshop on Mobile Robot Navigation Proceedings.*

Moravec, H. P. (1988), *Mind Children*, Harvard University Press, Cambridge, Massachusetts.

Nehmzow, U. & Smithers, T. (1991), "Mapbuilding using Self-Organising Networks in "Really Useful Robots"", in J.-A. Meyer & S. W. Wilson, eds, *Proceedings of the First International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, MIT Press, Cambridge, Massachusetts, pp. 152–159.

Neumann, J. V. & Morgenstern, O. (1944), *Theory of Games and Economic Behavior*, John Wiley and Sons, New York, New York.

Newcombe, F. & Ratcliff, G. (1989), Disorders of Visupspatial Analysis, in *Handbook of Neuropsychology, Volume 2*, Elsevier, New York, New York.

Newell, A., Shaw, J. & Simon, H. (1957), "Empirical Explorations with the Logic Theory Machine", in *Proceedings of the Western Joint Computer Conference 15*, pp. 218–329. Also appeared in Feigenbaum & Feldman (1963) and Luger (1995).

Newell, A., Shaw, J. & Simon, H. (1958), "Chess Playing Programs and the Problem of Complexity", *Journal of Research and Development* **2**, 320–335. Also appeared in Feigenbaum & Feldman (1963).

Newell, A., Shaw, J. & Simon, H. (1959), "A General Problem-Solving Program for a Computer", *Computers and Automation* **8**(7), 10–16.

Newell, A., Shaw, J. & Simon, H. (1961), *Information Processing Language V Manual*, Prentice Hall, Edgewood Cliffs, New Jersey.

Nicolis, G. & Prigogine, I. (1977), *Self-Organization in Nonequilibrium Systems*, Wiley, New York, New York.

Nilsson, N. J. (1965), *Learning Machines*, McGraw-Hill, New York, New York.

Nilsson, N. J. (1971), *Problem-Solving Methods in Artificial Intelligence*, McGraw-Hill, New York, New York.

Nilsson, N. J., ed. (1984), *Shakey the Robot*, Stanford Research Institute AI Center, Technical Note 323.

Payton, D. W. (1986), "An architecture for reflexive autonomous vehicle control", in *IEEE Robotics and Automation Conference*, San Francisco, California.

Payton, D. W. (1990), "Internalized Plans: A Representation for Action Resources", *Robotics and Autonomous Systems* **6**(1,2), 89–103. Also appeared in Maes (1990*b*).

Pylyshyn, Z., ed. (1987), *The Robot's Dilema: The Frame Problem in Artificial Intelligence*, Ablex Publishing Corp., Norwood, New Jersey.

Raibert, M. H. (1986), *Legged Robots That Balance*, MIT Press, Cambridge, Massachusetts.

Raibert, M. H., Brown, H. B. & Murthy, S. S. (1984), 3-D Balance Using 2-D Algorithms, in Brady & Paul, eds, *Robotics Research 1*, MIT Press, Cambridge, Massachusetts, pp. 215–224.

Ramachandran, V. S. & Anstis, S. M. (1985), "Perceptual Organization in Multistable Apparent Motion", *Perception* **14**, 135–143.

Roberts, L. G. (1963), Machine Perception of Three Dimensional Solids, Technical Report 315, MIT Lincoln Laboratory.

Rosenblatt, F. (1962), *Principles of Neurodynamics*, Spartan, New York, New York.

Rosenschein, S. J. & Kaelbling, L. P. (1986*a*), "The synthesis of digital machines with provable epistemic properties", in J. Halpern, ed., *Proceedings of the Conference on Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann Publishers, Los Altos, California, pp. 83–98.

Rosenschein, S. J. & Kaelbling, L. P. (1986*b*), "The Synthesis of Machines with Provable Epistemic Properties", in J. Halpern, ed., *Proc. Conf. on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann Publishers, Los Altos, California, pp. 83–98.

Rumelhart, D. E. & McClelland, J. L., eds (1986), *Parallel Distributed Processing*, MIT Press, Cambridge, Massachusetts.

Rumelhart, D., Hinton, G. & Williams, R. (1986), Learning Internal Representations by Error Propagation, in D. E. Rumelhart & J. L. McClelland, eds, *Parallel Distributed Processing*, MIT Press, Cambridge, Massachusetts, pp. 318–364.

Russell, S. J. (1989), "Execution Architectures and Compilation", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, Morgan Kaufmann, Detroit, Michigan, pp. 15–20.

Sacks, O. W. (1974), *Awakenings*, Doubleday, New York, New York.

Samuel, A. L. (1959), "Some Studies in Machine Learning Using the Game of Checkers", *Journal of Research and Development* **3**, 211–229. Also appeared in Feigenbaum & Feldman (1963) and Luger (1995).

Sarachik, K. B. (1989), "Characterising an Indoor Environment with a Mobile Robot and Uncalibrated Stereo", in *Proceedings IEEE Robotics and Automation*, Scottsdale, Arizona, pp. 984–989.

Sejnowski, T. & Rosenberg, C. (1987), "Parallel Networks that Learn to Pronounce English Text", *Complex Systems* **1**, 145–168.

Selfridge, O. G. (1956), "Pattern Recognition and Learning", in C. Cherry, ed., *Proceedings of the Third London Symposium on Information Theory*, Academic Press, New York, New York.

Shannon, C. E. (1950), "A Chess-Playing Machine", *Scientific American*.

Shortliffe, E. H. (1976), *MYCIN: Computer-based Medical Consultations*, Elsevier, New York, New York.

Simmons, R. & Krotkov, E. (1991), "An Integrated Walking System for the Ambler Planetary Rover", in *IEEE Proceedings on Robotics and Automation*, Sacramento, California.

Simon, H. A. (1969), *The Sciences of the Artificial*, MIT Press, Cambridge, Massachusetts.

Slagle, J. R. (1963), A Heuristic Program that Solves Symbolic Integration Problems in Freshman Calculus, in E. A. Feigenbaum & J. Feldman, eds, *Computers and Thought*, McGraw-Hill, New York, New York, pp. 191–206. From a 1961 MIT mathematics Ph.D. thesis.

Slate, D. J. & Atkin, L. R. (1984), Chess 4.5–The Northwestern University Chess Program, in P. Frey, ed., *Chess Skill in Man and Machine*, Springer-Verlag.

Smith, B. C. (1991), "The Owl and the Electric Encyclopedia", *Artificial Intelligence Journal* **47**, 251–288.

Steels, L. (1990), "Towards a Theory of Emergent Functionality", in *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, Massachusetts, pp. 451–461.

Steels, L. & Brooks, R., eds (1995), *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Sussman, G. J. (1975), *A Computer Model of Skill Acquisition*, Elsevier, New York, New York.

Teitelbaum, P., Pellis, V. C. & Pellis, S. M. (1990), "Can Allied Reflexes Promote the Integration of a Robot's Behavior", in *Proceedings of the First International Conference on Simulation of Adaptive Behavior*, MIT Press, Cambridge, Massachusetts, pp. 97–104.

Thorpe, C., Herbert, M., Kanade, T. & Shafer, S. A. (1988), "Vision and Navigation for the Carnegie-Mellon Navlab", *IEEE Trans. PAMI* **10**(3), 362–373.

Tinbergen, N. (1951), *The Study of Instinct*, Oxford University Press, Oxford, United Kingdom.

Tsuji, S. (1985), Monitoring of a building environment by a mobile robot, in Hanafusa & Inoue, eds, *Robotics Research 2*, MIT, Cambridge, Massachusetts, pp. 349–356.

Turing, A. M. (1937), "On Computable Numbers with an Application to the Entscheidungsproblem", in *Proceedings of the London Math. Soc. 42*, pp. 230–265.

Turing, A. M. (1950), "Computing Machinery and Intelligence", *Mind* **59**, 433–460. Also appeared in Feigenbaum & Feldman (1963) and Luger (1995).

Turing, A. M. (1970), Intelligent Machinery, in B. Meltzer & D. Michie, eds, *Machine Intelligence 5*, American Elsevier Publishing, New York, New York, pp. 3–23.

Turk, M. A., Morgenthaler, D. G., Gremban, K. D. & Marra, M. (1988), "VITS–A Vision System for Autonomous Land Vehicle Navigation", *IEEE Trans. PAMI* **10**(3), 342–361.

Uexküll, J. V. (1921), "Ummwelt und Innenwelt der Tiere". Berlin, Germany.

Ullman, S. (1984), "Visual Routines", *Cognition* **18**, 97–159.

Viola, P. A. (1989), "Neurally Inspired Plasticity in Oculomotor Processes", in *1989 IEE Conference on Neural Information Processing Systems–Natural and Synthetic*, Denver, Colorado.

Viola, P. A. (1990), Adaptive Gaze Control, Master's thesis, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Walter, W. G. (1950), "An Imitation of Life", *Scientific American* **182**(5), 42–45.

Walter, W. G. (1951), "A Machine That Learns", *Scientific American* **185**(5), 60–63.

Walter, W. G. (1953), *The Living Brain*, Duckworth, London, United Kingdom. Republished in 1961 by Penguin, Harmondsworth, United Kingdom.

Watkins, C. (1989), Learning from Delayed Rewards, PhD thesis, King's College, Cambridge, Cambridge, United Kingdom.

Waxman, A. M., Moigne, J. L. & Srinivasan, B. (1985), "Visual Navigation of Roadways", in *Proceedings of the IEEE Conference on Robotics and Automation*, St. Louis, Missouri, pp. 862–867.

Wehner, R. (1987), "'Matched Filters'–Neural Models of the External World", *Journal of Comparative Physiology* **A 161**, 511–531.

Wiener, N. (1948), "Cybernetics", *Scientific American* **179**(5), 14–19.

Wiener, N. (1961), *Cybernetics*, second edn, MIT Press, Cambridge, Massachusetts.

Wilkins, D. E. (1979), Using Patterns and Plans to Solve Problems and Control Search, Technical Report AI Memo 329, Stanford University.

Williams, M. R. (1983), "From Napier to Lucas", *Annals of the History of Computing* **5**(3), 279–296.

Wilson, D. (1966), "Insect Walking", Annual Review of Entomology, 11. 103–121.

Winograd, T. (1972), *Understanding Natural Language*, Academic Press, New York, New York.

Winograd, T. & Flores, F. (1986), *Understanding Computers and Cognition*, Addison-Wesley, Reading, Massachusetts.

Winston, P. H. (1972), The MIT Robot, in B. Meltzer & D. Michie, eds, *Machine Intelligence, 7*, John Wiley and Sons, New York, New York, pp. 431–463.

Winston, P. H. (1984), *Artificial Intelligence*, second edition edn, Addison-Wesley, Reading, Massachusetts.

Wood, W. B. (1988), "The Nematode *Caenorhabditis Elegans*", Cold Spring Harbor Laboratory. Cold Spring Harbor, New York.

Yap, C. (1985), Algorithmic Motion Planning, in J. Schwartz & C. Yap, eds, *Advances in Robotics, Vol. 1*, Lawrence Erlbaum Associates, Hillsdale, New Jersey.

Yarbus, A. (1967), *Eye Movements and Vision*, Plenum Press.