Autochthonous Behaviors - Mapping Perception to Action

Rod Grupen and Thomas C. Henderson*
COINS
University of Massachusetts, USA

Abstract

In this paper we describe an approach to high-level multisensor integration organized around certain egocentric behaviors. The task itself determines the sequence of sensing, the sensors used, and the responses to the sensed data. This leads to the encapsulation of robot behavior in terms of logical sensors and logical actuators. A description of this approach is given as well as some examples for dextrous manipulation and mobile robots.

*Department of Computer Science, University of Utah, USA. This author would like to thank Olivier Faugeras and the Robotics and Vision Group at INRIA (Sophia-Antipolis), France for the great sabbatical year there in 1988-89. This work was supported in part by NSF Grant IRI-8802585 and DARPA Contract DAAK1184K0017. All opinions, findings, conclusions or recommendations expressed in this document are those of the author.

1 Introduction

Figure 1 depicts the universe as a quantifiable state space. Within this space, subspaces are illustrated which represent the perceivable and actuatable spaces within the universe. The perceivable subspace represents those states which may be measured within the universe, the actuatable subspace represents those states which may be altered by employing a behavior of the system. In the discrete domain, the perceivable space represents states which are fully measureable by available (logical) sensors and the actuatable space are those states which may be transformed to other states by employing (logical) actuators. The dimensionality of the perceivable and actuatable subspace may not necessarily be the same. Moreover, both must be proper subsets of the universe. We will hypothesize an ecological niche in state space for such a system as the task subspace. These are a collection of states which, for whatever reason, identify crucial events in the universe. Such events in biological systems usually correspond to survial – such as the states abstractly indicative of food or danger.

We present such a depiction of an abstract system to discuss properties of systems which directly effect the theoretical limits of the transform from perception to action. Systems employing logical sensors and logical actuators whose state space description consists of disjoint perceivable and actuatable spaces do not have the ability – regardless of training, coaching, teaching, or otherwise cajoling – to effectively map perception to action. The measurable states, in this case, will not correlate to the states from which meaningful behavior can be derived. But, consider the case when the intersection of the perceivable and actuatable spaces is not empty. This condition suggests that certain properties of the universe can be measured, and that this perceived environment can be used to stimulate action such that perceivable state changes occur. Our system now has a region in state space which, in theory, permits abstract goals to be expressed. The proximity of these goals to perceived states can be determined and state changes can be selected which minimize the *distance* to the goal.

At this point in the discussion – rather than digressing into the role of random mutation and evolution in optimizing the task space niche of biological systems – we will instead turn our attention to the design of robotic systems. A (presumabley) human designer typically selects appropriate logical sensing and actuation mechanisms with which to express the desired task domain in an uncertain and partially unpredictable universe. We will call the process the design of *logical behaviors* and will draw on research in robotic control which is cognizant of the principles illustrated above.

Many approaches to multisensor integration have been proposed ranging from low-level descriptions of geometric data sensors[28,29,30] to high-level schemes[1,51,77]. Alternatively, one can focus on the sensors[48] or particular applications[2,6].

Our recent work has focused on a mid-level problem: the organization and integration of sensing in terms of intermediate level types of behavior – that is, activities which are not reflex, but which for the most part are not directly coupled to high-level "intelligent" behavior[45]. An example of such behavior is obstacle avoidance in a mobile robot. Here, data must be integrated from cameras, sonars, and perhaps other sensors as well. However, this function must be performed in an ongoing and automatic way. This is a learned behavior.

For the most part, the types of behavior involved are egocentric, i.e., they maintain spatio-temporal relations between the robot and the world. Our analysis is organized in terms of robot goals and behavior. This is accomplished by the use of what we call: **logical behaviors**. This approach allows for active control and integratoin of multisensor information in the framework of a specific task. We provide examples of the application of these ideas to impedance control, dextrous manipulation and mobile robots.

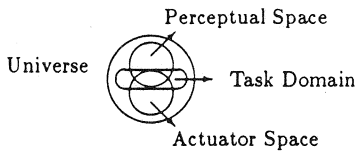


Figure 1. The Task-Perceptor-Actuator Trilogy

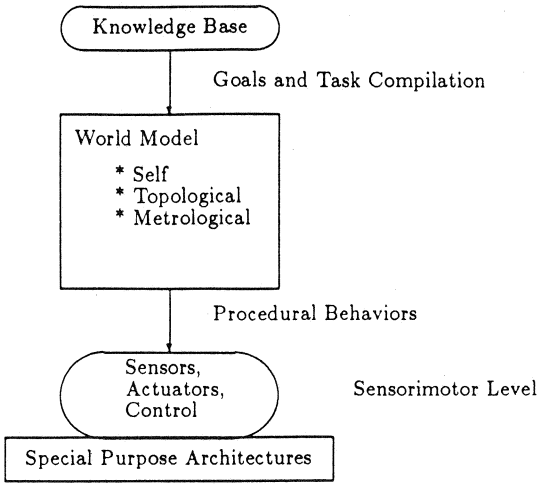


Figure 2. Autonomous Robot Research

2 Background

Multisensor integration has received a good deal of attention in recent years due to the availability of sensors, actuators, and processors. Two major testbeds for such work are:

- robotic workcell automation, and
- mobile autonomous robots.

The first of these involves applying strong knowledge-based techniques to the manufacturing environment, while the second concerns integrating several levels of information processing into a single autonomous system. We restrict our attention here to dextrous manipulation and mobile robots.

Autonomous mobile robots have been studied in a wide range of contexts. Figure 2 imposes an organization on most of the typical keywords. Obviously, the problem of navigation is basic to mobile robots and consequently has been studied by many people on specific implementations [20,22,40,44, 54,59,70,84,85,89,92]. Most such systems must use sensors (e.g., sonar or cameras [26,31,33,91]) and actuators and must control them [32,52,69,75,94,95]. The use of sensors requires the study of uncertainty management [43,78,83,86] and multisensor integration [21,30,50,65,67,72]. More global approaches to the sensorimotor problem can be found in [1,11,39], and special purpose architectures are being planned [4,88,97].

One level up, the mapping of procedural behaviors onto the sensorimotor control structure is of interest [3,5,14,46,47,48,49,93]. The world representations also exist at this level: both the metrological [7,8,73,80], where precise measurement is paramount, and topological [10,12,19,35,36,37,38,41,55,63,68, 87,90,98], where adjacency relations are useful for path planning, etc. It is even possible to study primitive forms of learning in this context [82,90].

Broader studies are usually oriented towards particular applications (e.g., the nuclear industry [17, 96], road following [23,24,74]) or towards well-defined, but limited goals (e.g., indoor [13,34] or outdoor [60,64] navigation).

Finally, the 'highest' level involves the specification and representation of the knowledge appropriate to a given task [18,57,58,62,76] and its compilation into executable robot behavior (or programs) [27,

51,56,61]. The literature is quite large on most of these subjects, and these references are intended as representative of the work in this area. It should be pointed out that most system designers use a central blackboard and some form of direct production system or a compiled version (i.e., a decision tree) to represent knowledge.

From this short summary, it can be seen that the scope of autonomous robot research is indeed vast, but the difficult problems found here are yielding to the steady advance of technical and theoretical developments. In the remainder of this paper, we describe current work on the mobile autonomous robot at INRIA.

3 Behavior Based Sensing and Control

In the most general sense, a robot interacts with its environment by applying operators to the perceived state of the environment. The state and operator may be cognitive - effecting the composition of state parameters without physically altering the environment; on the other hand, elements of the robot's surface may actually be applied to the geometry of the environment. In the latter case, the contacts may be derived from the robot's wheels or bumpers in the case of a mobile cart, or from the fingertips, proximal phalanges, palm or arm of a dextrous manipulator. Characteristics of the environment, the task, and the robot kinematics may be used to construct goal oriented behaviors.

The development of controllers for robotic systems is typically a generalization of the approach used in low level feedback controllers. Elements of the system state are measured and used to quantify the error of the system with respect to a desired state. The operation of the system tends to reduce the state error to zero. The nature of the feedback variables determines the nature of the response. Adaptable control schemes can optimize the response over uncertain inputs by varying the weighting of the feedback variables; however, types of behaviors which are not defined *a priori* cannot be expressed. This suggests that a single control law is not sufficient to manage the complexity of the general purpose robot systems. Control methodologies have been developed which partition the state space of complex systems into disjoint regions, each with an associated control law [79]. The operation of these systems is represented by a finite state automata where state transitions are triggered by sensory events. This approach produces sequences of behaviors in the system.

Behavior based control schemes generalize this approach. Elemental behaviors are instantiated which span the problem domain (see Braitenberg[14]). Braitenberg's work was the precursor of many similar systems, including the subsumption architecture proposed by Brooks[15]. The subsumption architecture is an approach which was developed to construct systems which require composite behaviors[16,15]. Concurrent control laws are defined, each of which acquiring the sensory data necessary for that particular behavior. The so-called *activity producing subsystems* are integrated in a hierarchy in which primary behaviors reside at the lowest levels. Higher level behaviors are used to modulate the output produced by low level behaviors. The work demonstrated a hardwired system tuned to perform a particular task, the navigation of an autonomous vehicle.

These approaches are generalized in the society structures proposed by Minsky[71]. This proposed structure is motivated by observed problem solving behaviors in humans. Agencies are postulated which serve as *proto-specialists* over limited problem domains. The society of such agents is capable of goal oriented behaviors with dynamic priorities based on the current state of the composite system.

The logical behavior systems proposed in this paper are based on a similar perspective. Independent, elemental behaviors are defined which span the required problem domain. We generalize the notion of a behavior to any process which maps information abstracted from (logical) sensors to state transitions which may be mapped onto (logical) actuators. Once again, a logical sensor need not be linked directly or indirectly to a physical sensor, but may represent any hypothetical state from which a state transition is desired. Likewise, the logical actuator need not employ a DC motor, for example, but will transform a state in the actuatable space to some other state. This property provides a mechanism for cognitive and reflexive mappings from sensors to actuators. We also note that the distinction between planning and execution is in some sense a function of whether the logical sensor is in fact terminated at a physical sensor, and whether the logical actuator terminates at a physical actuator.

We describe below the application of this approach to dextrous manipulation and to mobile robots. The example of the design of logical behaviors for multifingered manipulator control includes complex kinematics, multi-functional mechanisms, and complex tasks. The other application is the design of an obstacle avoidance behavior for a mobile robot. We will make an effort to keep our primary focus on the even larger problem domain describing general transformations from sensors to actuators.

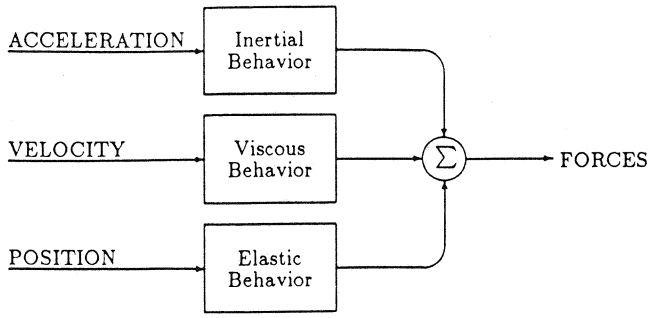


Figure 3. Three Logical Behaviors Which Constitute the Impedance Controller

4 Logical Behaviors for Generalized Impedance Control

Drake designed a passively compliant device which automatically compensates for uncertainty in certain classes of assembly operations[25]. Salisbury employs a stiffness controller to support the construction of stable grasps[81], and Lozano-Pérez *et al.* discuss the use of the generalized damper to plan fine motion assembly strategies[66]. These instances of manipulator control map sensor data to action by modeling the manipulator as an impedance relative to an environmental admittance. As such, the manipulator measures deviations from nominal positions or velocities and applies a correcting force.

To illustrate the use of logical behaviors for multisensor integration, consider an implementation of Hogan's impedance control[53]. Hogan argues that to:

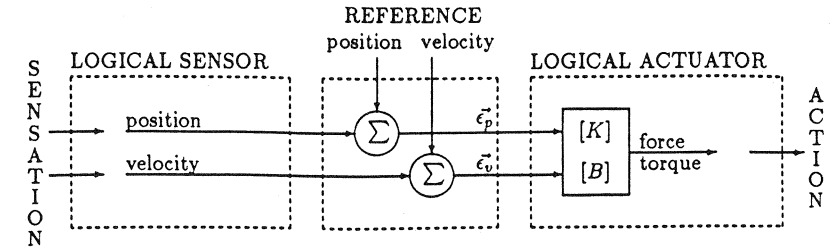
> ensure physical compatability with the environmental admittance, something has to give, and the manipulator should assume the behavior of an impedance.

Others have noted the usefulness of impedance control - loosely defined for our purposes as behaviors which map errors in position, velocity, or acceleration to forces. The terms in the impedance controller are linearly independent functions of separable state variables, $\vec{x}, \dot{\vec{x}}$, and $\ddot{\vec{x}}$. Figure 3 illustrates how the impedance controller may be considered to be a superposition of three separate impedance behaviors: an inertial behavior, a viscous impedance behavior and an elastic impedance bhavior.

The remainder of the presentation will consider only the visco-elastic components of the generalized impedance controller. The structure of each of these logical behaviors consists of a logical sensor, an (optional) reference input, and a logical actuator. The logical sensor is any combination of hardware and software which measures and/or hypothesizes the state of the system[48]. The logical actuator is likewise, any combination of hardware and/or software which transforms the state representation. The logical actuator may simply transform the abstracted state representation, or it may actually employ hardware actuators to physically change the state of the system. To understand the utility in an abstract notion of the logical impedance behaviors, consider the various incarnations of the visco-elastic impedance controllers presented in Figure 4.

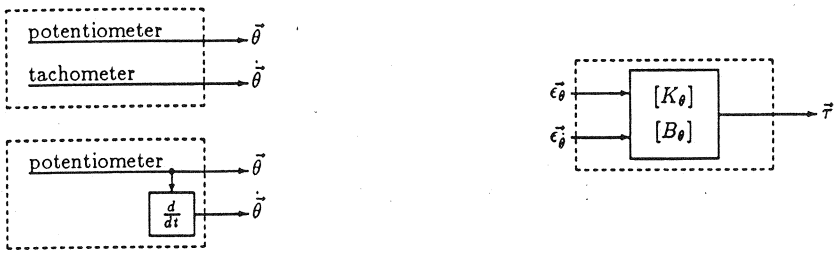
The figure defines the constraints which govern the construction of a logical behavior. In this case the behavior describes a transformation from the position/velocity domain into the force/torque domain. The behavior is represented as a combination of a logical sensor and a logical actuator. The generic logical behavior on the top of Figure 4 defines the data type consistency which must be maintained during the construction of the logical impedance behavior. In essence, all data entering or leaving the summation block in Figure 4 must be consistent. We have thus defined the type of the: characteristic output vector (cov) for the logical sensor, the reference input vector (riv), and the characteristic input vector (civ) of the logical actuator.

Figure 4(a) depicts the commonplace joint impedance controller. We illustrate two logical sensors which yield the joint space position and velocities required. The logical actuator simply computes



LOGICAL SENSORS LOGICAL ACTUATORS

(a) Visco-Elastic Joint Position Servoes



(b) Visco-Elastic Cartesian Endpoint Control

(c) Visco-Elastic Object Pose Control

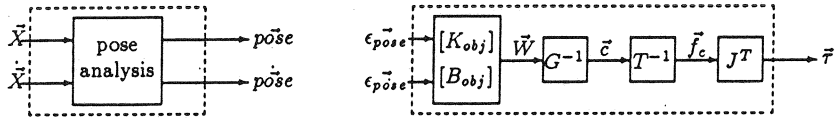


Figure 4. Implementations of Logical Impedance Behaviors

torques which suppr
two commonly used
presented in Figure
impedance paramete
somewhat different l
and a logical actuat
Cartesian pose impe
the set of Cartesian
The corresponding l
restoring wrenches,
well. The inverse of
applied through each
wrenches, such that:

then,

and, the force comm

Finally, the forces ap
appropriate manipul

Beyond the relat
is an instance of the
sensor and logical ac
multisensor control.

5

A primary motiv
abstraction that is p
will first suggest sev
manipulation. The r
a basis for autonom
impedance behavior
that follows. These l
[42].

The **wrench clo**
the equilibrium posit

1. the geometry o
2. the type(s) of c
   and
3. the pose of the

This *sensor* data ma
knowledge of the ma
is expressed analytic
of the object is know
in wrench space whi
object's six degrees
the logical actuators
by manipulator cont
descent in the error s

The **isotropic m**
the wishes of the wr

torques which suppress errors in joint space. Figure 4(b) presents logical behavior expressions of two commonly used Cartesian endpoint controllers. The second simply employs the logical sensors presented in Figure 4(a) and a logical actuator which suppresses joint space errors using Cartesian impedance parameters as suggested by Salisbury[81]. The first Cartesian controller shown employs a somewhat different logical sensor which expresses the position and velocity directly in Cartesian space and a logical actuator which supresses errors in Cartesian space. Figure 4(c) presents an object frame Cartesian pose impedance behavior. Another stage is added to the logical sensor which transforms the set of Cartesian contact positions and velocities into an object frame pose and velocity in 6-space. The corresponding logical actuator maps errors in the pose parameters relative to the reference into restoring wrenches, $\vec{W}$. This wrench is in n-space and contains internal (null space) components as well. The inverse of the Grip Jacobian is used to map the restoring wrench into individual wrenches applied through each contact location. If the transform T is defined to map contact forces into contact wrenches, such that:

$$\vec{w}_i = T\vec{f}_i$$

then,

$$\vec{f}_i = T^{-1}\vec{w}_i$$

and, the force commands at each contact are expressed as:

$$\vec{f}_i^c = T^{-1}(\vec{c}^{\,T}\vec{w}_i).$$

Finally, the forces applied at each contact location are transformed into actuator torques by way of the appropriate manipulator Jacobian.

Beyond the relative complexity of the impedance controllers presented in Figure 4, each controller is an instance of the same logical behavior. Each instance represents an effective assemblage of logical sensor and logical actuators for some combination of task objectives and constraints in the context of multisensor control.

5 Logical Behaviors for Grasping and Manipulation

A primary motivation for developing the logical behavior formalism is the modularity and data abstraction that is provided in the design of complex robotic controllers. To illustrate this feature, we will first suggest several behaviors which are understood to be useful in the context of grasping and manipulation. The resulting behaviors may be used as a programming language for manipulation, or as a basis for autonomous behavior composition. These behaviors are in fact supported by the Cartesian impedance behaviors described in Section 4. We will employ two principal behaviors in the discussion that follows. These behaviors and others useful in grasping and manipulation are described in detail in [42].

The **wrench closure behavior** is defined to construct 6 dimensional constraint envelopes about the equilibrium position of the object. The logical sensors for this behavior must determine:

1. the geometry of the graspable surface,
2. the type(s) of contact(s) delivered to the object's surface by the manipulator and the environment, and
3. the pose of the object.

This *sensor* data may be derived from a variety of physical sensors, or derived from basic principles and knowledge of the manipulator and the object. In the simulations presented, the geometry of the object is expressed analytically, the contact type is uniformly a point contact with friction model, and the pose of the object is known a priori. The reference input to the logical behavior is a 6-dimensional volume in wrench space which expresses both positive and negative sense wrench magnitudes in each of the object's six degrees of freedom. This *goal* represents a stable grasp objective. The error submitted to the logical actuators represents the difference between the current state of the wrench volume generated by manipulator contacts and the reference input. The logical actuator in this case performs a gradient descent in the error space toward suitable wrench closure states.

The **isotropic manipulator behavior** is designed to condition the manipulator to be compliant to the wishes of the wrench closure behavior. This behavior employs a logical sensor to compute a scalar

manipulability index for the hand. This state description is derived from the kinematic configuration of all the fingers in a grasp. The scalar index is normalized so that the reference goal for this behavior is implicitly to generate a unity index for the hand. This behavior performs a gradient ascent in the manipulability index toward isotropic hand states.

Logical Sensors	*Logical Actuators*	
Object Geometry	Closure Gradient Descent	contact positions
Contact Positions	Hand Index Gradient Ascent	position/orientation

Examples are presented for cylindrical and spherical test objects. All examples employ the Utah/MIT hand geometry. A top view and a side view of the hand/object system are presented. Intermediate positions for the hand frame y- and z-axes are shown. The initial hand frame position is identified in bold print; for clarity, only the final finger configuration is shown. Typical computation time for the four fingered grasps is approximately 10 *seconds* of CPU time on a VAX 750.

The original task stack submitted to the system for the initial grasp task is illustrated in Figure 5.

```
type:               approach
screw task:         NA
contact set:        all fingers
nomadic contacts:   all fingers
↓
type:               condition
screw task:         hypercubic wrench volume
contact set:        all fingers
nomadic contacts:   all fingers
↓
NIL
```

Figure 5. The Task Stack Representation of the Initial Grasp Task

Figures 6, 7 and 8 present the grasps of the cylinder produced when 2, 3 and 4 fingertip contacts are applied to the task, respectively. In Figure 8, the third finger (the ring finger) does not oppose the thumb, but goes to a position where it more effectively complements the wrench subspaces produced by the other contacts. In (b), the task is modified dynamically by adding an incremental task which is the wrench subspace produced by the index finger. The trajectories of the other fingers are then tethered elastically to the index finger. The fingers are essentially grouped into a virtual finger. Which grasp is better for the robot hand is not clear, but (b) appears more anthropomorphic.

Figures 9, 10 and 11 present the grasps of a sphere produced when 2, 3 and 4 fingertip contacts are applied to the task, respectively. In Figure 11 the super symmetric object and the functionally redundant manipulator produced convergence difficulties in the geometry synthesis. There appear to be closely spaced meta-stable states in the contact geometry solution. The convergence problem was controlled by designating a virtual finger over the middle and ring fingers of the hand; the result is not intuitively satisfying. This example suggests the application of additional constraints in the form of virtual fingers and/or geometrical restrictions limiting the portion of the object surface which may be used to address a particular task.

6 Logical Behaviors for Obstacle Avoidance

In this section we consider the following problem: suppose that our mobile robot is wandering through an unknown indoor environment. The robot must:

- **incrementally build a 3-D representation of the world** (i.e., determine its motion and integrate distinct views into a coherent global view),
- **account for uncertainty in its description** (i.e., explicitly represent, manipulate and combine uncertainty), and
- **build a semantic representation of the world** (i.e., discover useful geometric or functional relations and semantic entities).

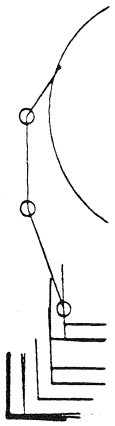


Figure 6. A Two Fi
Volume in Six DOF

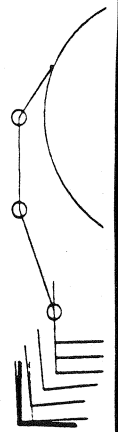


Figure 7. A Three
DOF

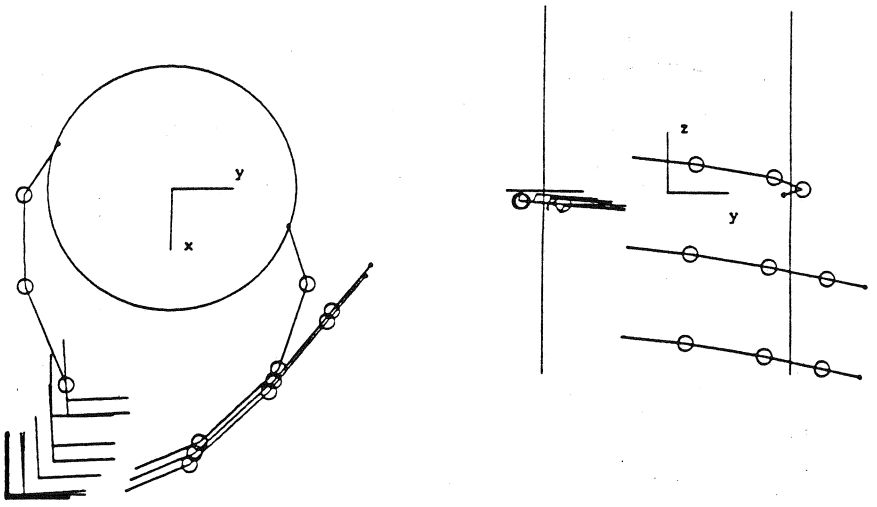


Figure 6. A Two Fingered Grasp on a Cylinder 4 *inches* in Diameter; the Task is a Hypercubic Wrench Volume in Six DOF

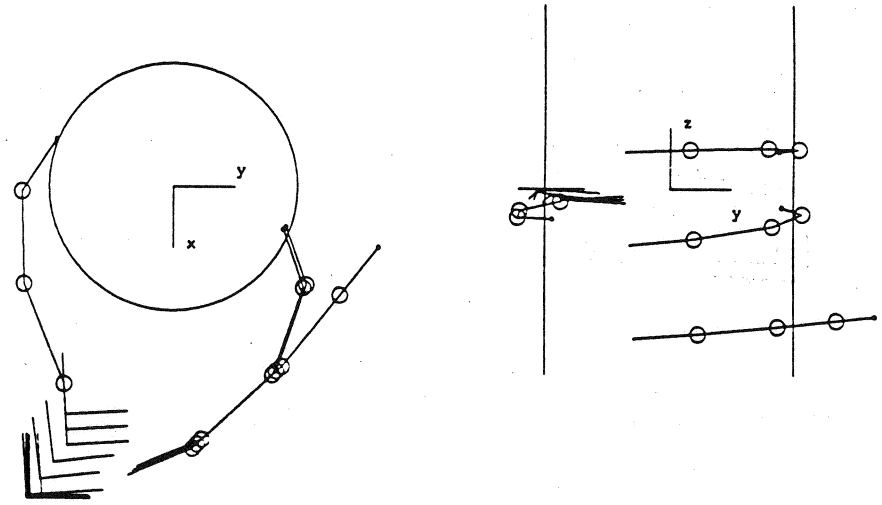


Figure 7. A Three Fingered Grasp on the Cylinder; the Task is a Hypercubic Wrench Volume in Six DOF

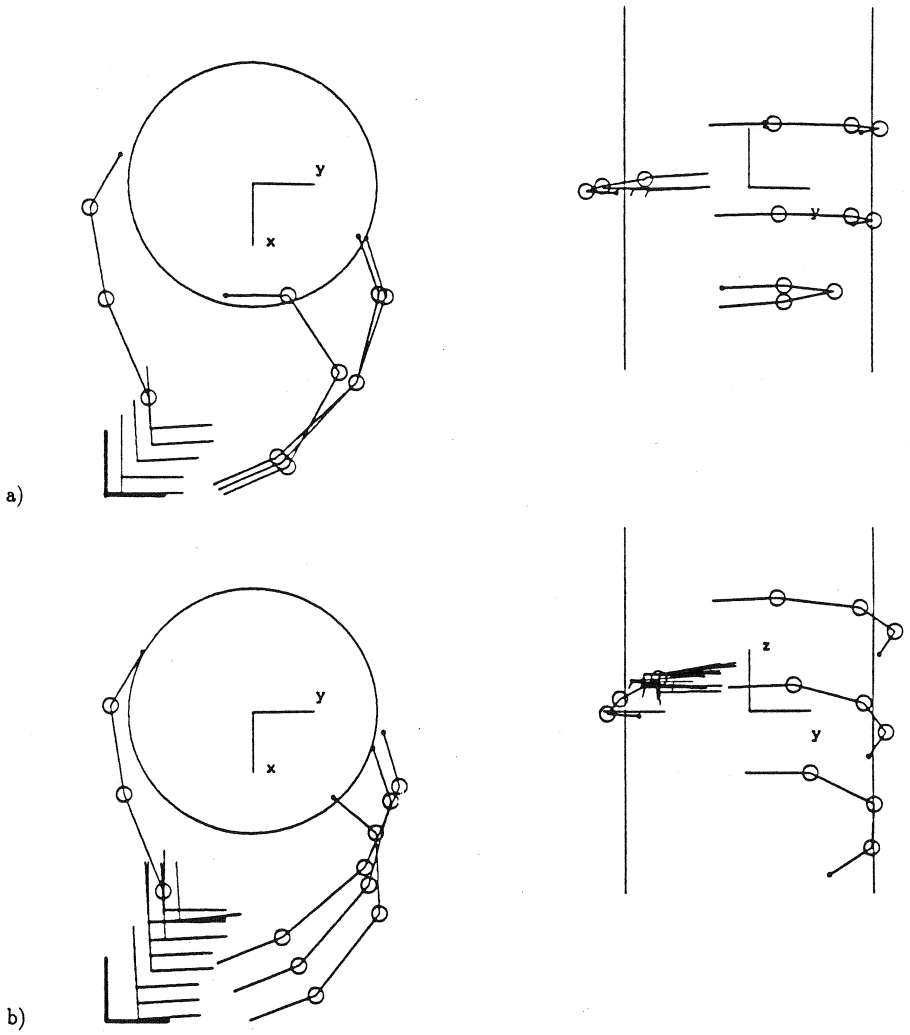


Figure 8. (a) A Four Fingered Grasp on the Cylinder for a Hypercubic Wrench Volume Task; (b) the Same Task with the Virtual Finger Designation over the Index, Middle and Ring Fingers

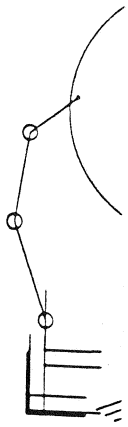


Figure 9. A Two F
Volume in Six DO

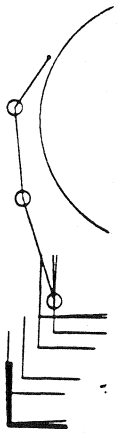


Figure 10. A Three
DOF

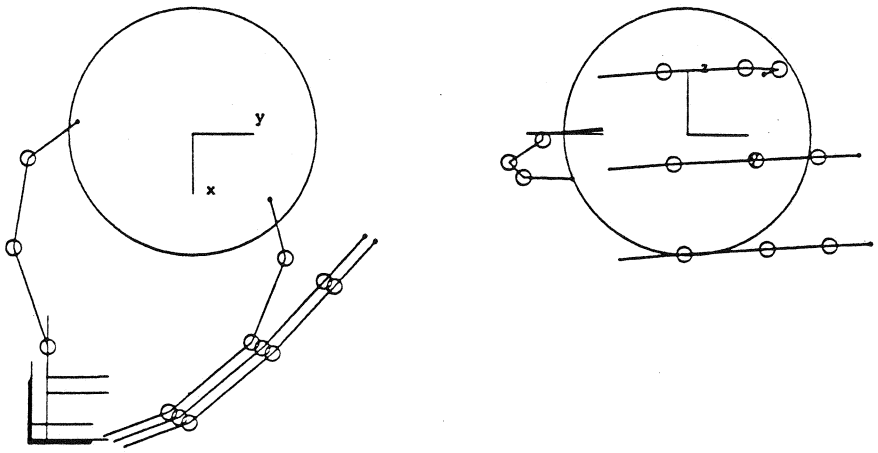


Figure 9. A Two Fingered Grasp on a Sphere 4 *inches* in Diameter; the Task is a Hypercubic Wrench Volume in Six DOF

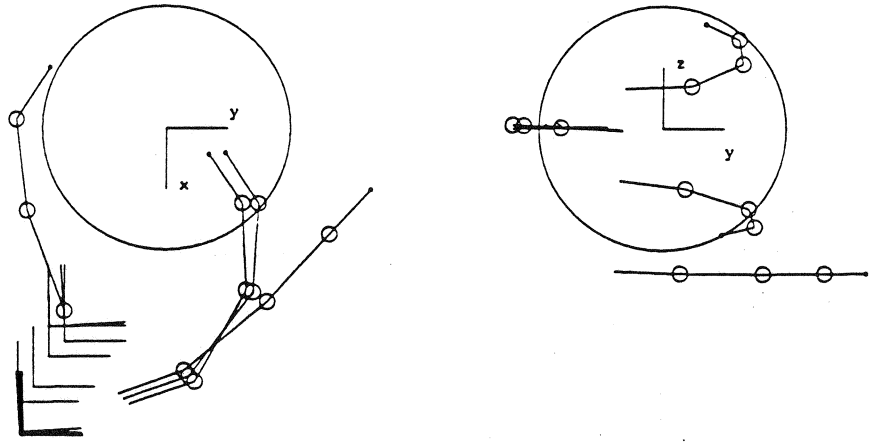


Figure 10. A Three Fingered Grasp on the Sphere; the Task is a Hypercubic Wrench Volume in Six DOF

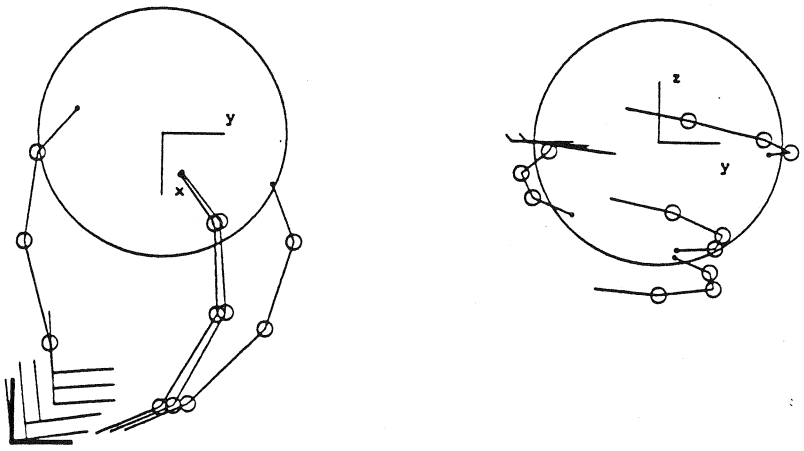


Figure 11. A Four Fingered Grasp on the Sphere

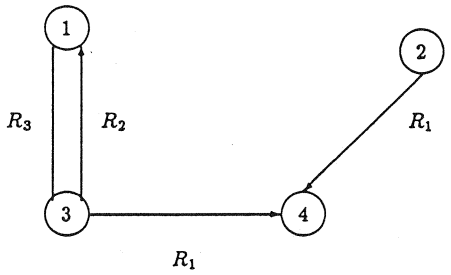


Figure 12. Semantic Net Defining World Model

Here we describe an approach to solving the third problem. (See [6] for details on efficient techniques for producing a local 3-D map from stereo vision and structure from motion as well as a method for combining several viewpoints into a single surface and volume representation of the environment and which accounts for uncertainty.)

The mobile robot must use the 3-D representation to locate simple generic objects, such as doors and windows, and eventually more complicated objects like chairs, desks, file cabinets, etc. The robot can then demonstrate task-based behavior such as going to a window, finding a door, etc. The representation should contain semantic labels (floor, walls, ceiling) and object descriptions (desks, doors, windows, etc.).

The proposed approach is straightforward and exploits our previous work on logical sensors, the Multisensor Knowledge System, and multiple semantic constraints. The World Model is defined in terms of a semantic network (e.g., see Figure 12). The nodes represent physical entities and the relations are (currently) geometric. "Behind" each node is a logical sensor which embodies a recognition strategy for that object. The relations are simply tabulated.

A goal for the robot is defined by adding a node representing the robot itself and relations are added as requirements (see Figure 13). This method permits the system to focus on objects of interest and to exploit any strong knowledge that is available for the task. The added relations are satisfied (usually) by the robot's motion.

As an example, consider the world model in Figure 14 which represents a specific office. The addition of the robot and the "Next_to" relation fires the "Find_door" logical sensor. This in turn causes the strategy for door finding to be invoked. Such a strategy may attempt shortcuts (quick image cues) or may cause a full 3-D representation to be built and analyzed. Logical behaviors are then the combined

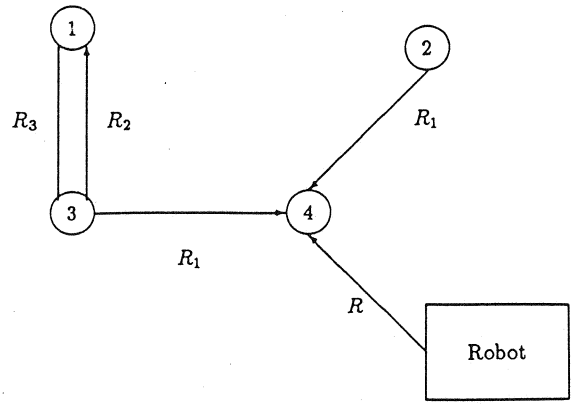


Figure 13. Defining the Robot Task

logical sensors and motion control required to satisfy the "Next_to" relation.

Note that it is in the context of such a strategy that high-level multisensor integration occurs in goal-directed behavior. We are currently implementing a testbed for experimentation.

6.1 Robot Behavior as Real-time Programming

Robots must maintain a permanent interaction with the environment, and this is the essential characteristic of *reactive programs*. Other examples include real-time process controllers, signal processing units and digital watches.

We have selected the Esterel synchronous programming language[9] as the specification language for the reactive kernel of the robot's behavior. A reactive system is organized in terms of three main components:

- **reactive kernel**: specified in Esterel and compiled into C or CommonLisp for execution,
- **interface code**: handles drivers and sensors, and
- **process or data handling code**: routine calculations.

The programs produced are:

- **deterministic**: produce identical output sequences for identical input sequences,
- **concurrent**: cooperate deterministically, and
- **synchronous**: each reaction is assumed to be instantaneous.

Interprocess communication is done by instantly broadcasting events, and statements in the language take time only if they say so explicitly; for example:

every 1000 MILLISECOND **do emit** SECOND **end**

In this example, a SECOND signal is sent every thousand milliseconds.

Thus, Esterel provides a high-level specification for temporal programs. Moreover, the finite state automata can be analyzed formally and give high performance in embedded applications. They help encapsulate the specification of sensing and behavior from implementation details. This simplifies simulation, too.

Other advantages include the fact that synchrony is natural from the user's viewpoint; e.g., the user of a watch perceives instant reaction to pushing a control button on the watch. Synchrony is also natural to the programmer. This reconciles concurrency and determinism, allows simpler and more rigorous

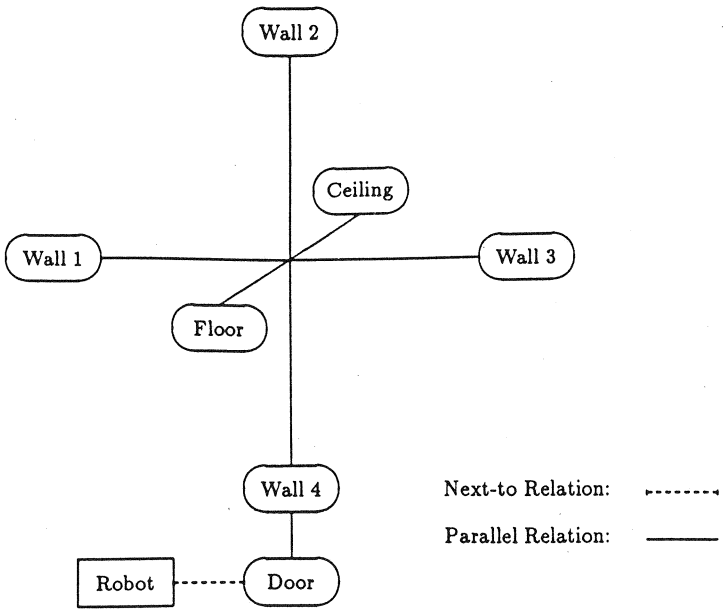


Figure 14. A Representation of the Command: "Go to the office door"

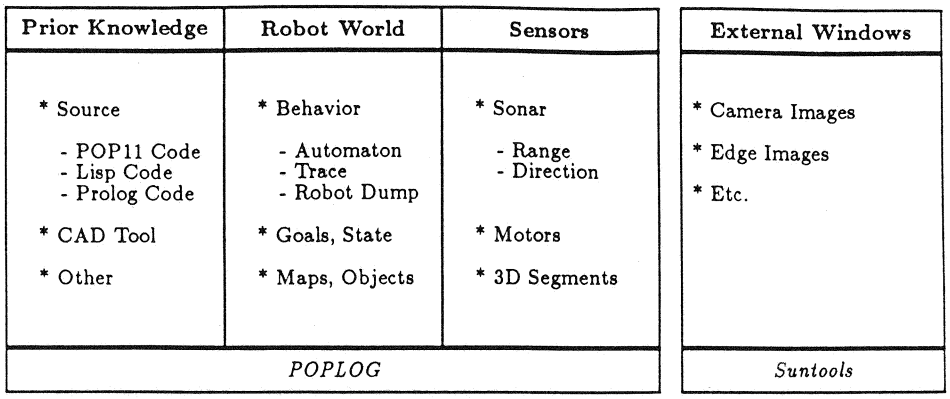


Figure 15. The Debugging System Organization

programs and separates logic from implementation. Finally, such automata are easily implemented in standard programming languages.

Details of the language are not given here; however, a brief summary is in order:

- **variables**: not shared; local to concurrent statements.
- **signals**: used to communicate with environment or between concurrent processes; carry staus (present or absent) and value (arbitrary type).
- **sharing law**: instantaneous broadcasting; within a reaction, all statements of a program see the same status and value for any signal.
- **statements**: two types:
  1. standard imperative style, and
  2. temporal constructs (e.g., **await** *event* **do**).

An extremely useful output from Esterel is a verbose description of the automaton. This can be used for debugging purposes. Esterel also produces a C program which implements the automaton.

Another useful output is a CommonLisp version of the automaton. This makes simulation straightforward, so long as reasonable functions can be written which simulate the world and the physical mechanisms of the robot. But these, too, can be specified in Esterel and then combined.

6.2 Robot Behavior Debugging Environment

In developing Esterel specifications for robot behavior and sensor control, we are faced with the problem of integrating diverse kinds of knowledge and representations. In particular, debugging robot behaviors requires knowledge of the world model, the robot's goals and states, as well as the behavior specification, and sensor data (intensity images, sonar data, 3-D segments, etc.). Figure 15 shows the current implementation organization. We use POPLOG (an interactive environment which combines CommonLisp, Prolog and Pop11) to support manipulation and display of prior knowledge, the robot world, and some sensor data, while other Suntool-based utilities support display of the trinocular stereo camera images, etc.

Figure 16 shows a representative collection of windows which provide:

- POPLOG source code (window management, etc.)
- Prolog source (semantic entity definition; e.g., walls, doors, etc.)
- sensor data display (e.g., sonar range data, 3D segments)
- Esterel generated automaton (e.g., COMBINE.debug)

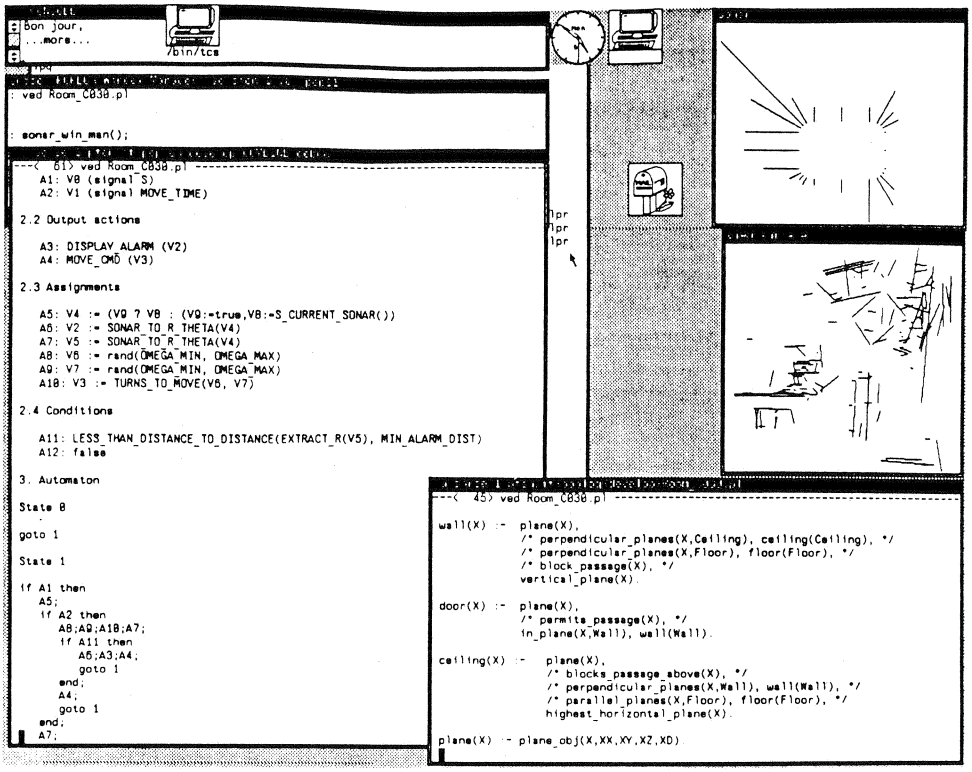


Figure 16. Collection of Windows for Debugging

Esterel permits state tracing during execution, and this combined with access to the robot's sensory data permits rapid and accurate debugging. In Appendix A we give the details for the specification of a wandering robot which must avoid colliding with objects in the world. This specification has been compiled and loaded onto the robot and successfully executed.

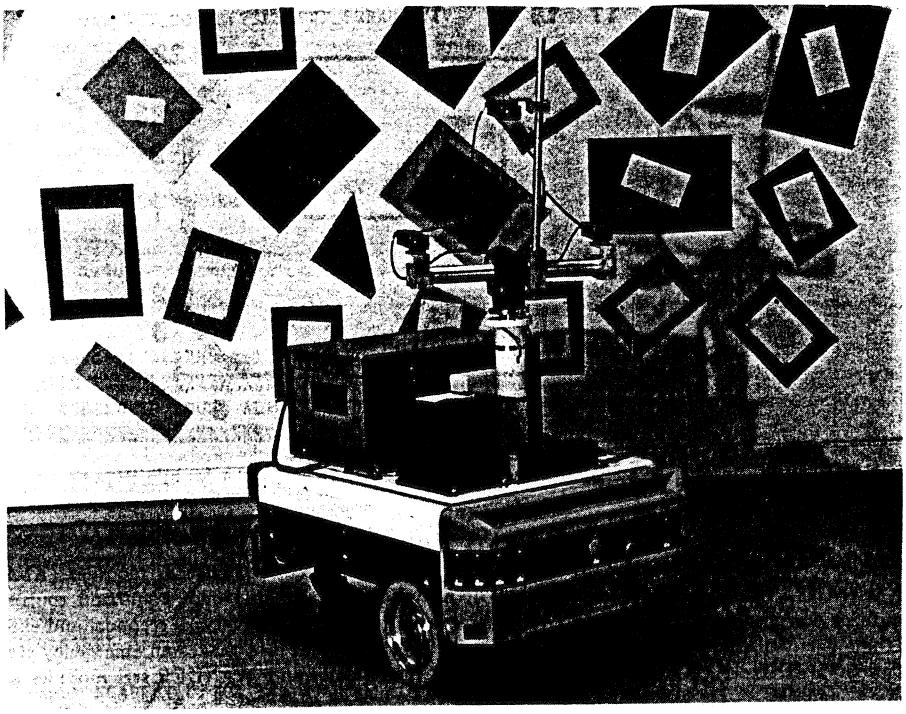


Figure 17. The INRIA Mobile Robot

6.3 Mobile Robot

There is an operational mobile robot at INRIA (Sophia-Antipolis). It is similar to other mobile robots (e.g., like those at CMU or Hilare at LAAS). Figure 18 shows the geometry of the robot (length: 1.025m, width: .7m, and height: .44m) and the locations of the sonar sensors. The two rear wheels drive the robot.

The onboard processing consists of two M68000 series microprocessors on a VME bus; one controls the sonar sensors, and the other runs the real-time operating system, Albatros. The two main wheels are controlled separately, and the system has an odometer.

High-level multisensor integration must be investigated in the context of real-world problems. We have described current work on an autonomous mobile vehicle under development at INRIA. We propose "logical behaviors" as an approach to robot goal representation and achievement.

We intend to continue development of algorithms, architectures and systems for multisensor robotic systems. Moreover, we are currently investigating the simulation of such systems; this involves embedding the reactive kernel in a modeled robot world. Finally, as can be seen by the rough nature of the definitions of walls, doors, etc., we must develop a suitable formal model of the world in which the robot finds itself. We intend to exploit optimized refinements of conceptual clusters defined in first order predicate calculus.

7 Summary

We have proposed **logical behaviors** as a technique for organizing multisensor integration and control. Such an approach allows encapsulation of sensing and actuation and relates those activities to a specific task. Several examples have been presented ranging from impedance control for a robot manipulator to obstacle avoidance for a mobile autonomous robot. We are currently extending the environment to include better debugging and simulation environments.

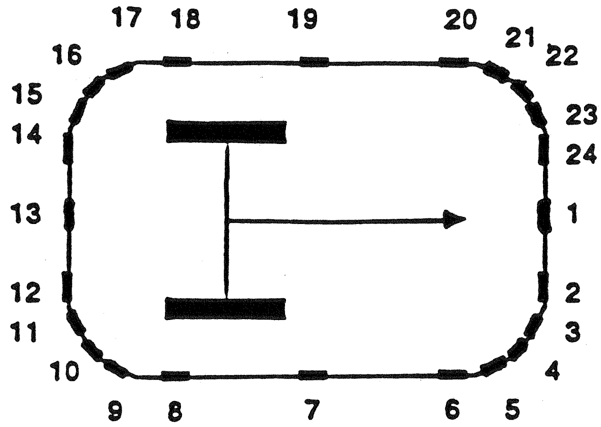


Figure 18. The Geometry and Sensor Placement on the INRIA Mobile Robot

A Wandering Robot Example

In this appendix, a system is developed which combines several ESTEREL modules (ALARM, GET_MIN_DISTANCE, WANDER and COMBINE) with the on-board robot command routines to generate random robot movement. The robot generates a random move every 10 seconds, and executes it; if there is any obstacle closer than the predefined threshold, the robot makes an emergency stop.

The COMBINE.strl, ALARM.strl, GET_NEAREST_OBJ.strl and WANDER.strl modules are as follows:

```
% $Header: COMBINE.strl,v 1.1 88/12/07  tch Locked $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  MODULE TO COMBINE READING WITH WATCHING THE SONAR SENSORS%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

module COMBINE:

type DISTANCE,
     PING,
     R_THETA,
     MOVE;

constant OMEGA_MIN, OMEGA_MAX : integer;

input S;
input MOVE_TIME;

relation MOVE_TIME => S;

sensor CURRENT_SONAR (PING);

output DISPLAY_ALARM(R_THETA);
output MOVE_CMD(MOVE);

[
signal GET_SONAR(PING), NEAREST_OBJ(R_THETA) in

  every S do
```

```
    emit GET_SONAR(?CURRENT_SONAR)
  end
||
  copymodule ALARM
||
  copymodule GET_MIN_DISTANCE
||
  copymodule WANDER
end
]
.


% $Header: ALARM.strl,v 1.1 88/12/07 tch Locked $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  MODULE TO SOUND ALARM IF OBJECT TOO CLOSE       %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

module ALARM:

type DISTANCE,
     PING,
     R_THETA;

constant MIN_ALARM_DIST : DISTANCE;

input NEAREST_OBJ(R_THETA) ;
input GET_SONAR(PING) ;

output DISPLAY_ALARM(R_THETA);

function LESS_THAN_DISTANCE_TO_DISTANCE(DISTANCE,DISTANCE) : boolean;
function EXTRACT_R(R_THETA) : DISTANCE;
function SONAR_TO_R_THETA(PING) : R_THETA;

every immediate NEAREST_OBJ do
  if LESS_THAN_DISTANCE_TO_DISTANCE(EXTRACT_R(?NEAREST_OBJ),MIN_ALARM_DIST)
    then emit DISPLAY_ALARM(SONAR_TO_R_THETA(?GET_SONAR))
  end
end
.

% $Header: GET_MIN_DISTANCE.strl,v 1.1 89/1/17 tch Locked $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  MODULE TO GET MINIMUM DISTANCE FROM SONARS      %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

module GET_MIN_DISTANCE:

type PING,
     R_THETA;

input GET_SONAR(PING) ;

output NEAREST_OBJ(R_THETA);
function SONAR_TO_R_THETA(PING) : R_THETA;
```

```
every immediate GET_SONAR do
  emit NEAREST_OBJ(SONAR_TO_R_THETA(?GET_SONAR))
end
.

% $Header: WANDER.strl,v 1.1 88/12/22 tch Locked $
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%  MODULE TO GENERATE RANDOM MOVES          %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

module WANDER:

type MOVE;

constant OMEGA_MIN, OMEGA_MAX : integer;

input MOVE_TIME;

output MOVE_CMD(MOVE);

function rand(integer, integer) : integer;
function TURNS_TO_MOVE(integer, integer) : MOVE;

every MOVE_TIME do

 var left_wheel_turns, right_wheel_turns : integer in

  left_wheel_turns := rand(OMEGA_MIN, OMEGA_MAX);
  right_wheel_turns := rand(OMEGA_MIN, OMEGA_MAX);
  emit MOVE_CMD(TURNS_TO_MOVE(left_wheel_turns,right_wheel_turns))
 end
end
.
```

The finite state machine produced for COMBINE is:

```
                    Automaton COMBINE  (Debug Format)

1. Memory allocation

   V0: boolean (boolean of signal S)
   V1: boolean (boolean of signal MOVE_TIME)
   V2: R_THETA (value of signal DISPLAY_ALARM)
   V3: MOVE (value of signal MOVE_CMD)
   V4: PING (value of signal GET_SONAR)
   V5: R_THETA (value of signal NEAREST_OBJ)
   V6: integer (variable left_wheel_turns)
   V7: integer (variable right_wheel_turns)
   V8: PING (value of sensor CURRENT_SONAR)
   V9: boolean (boolean of sensor CURRENT_SONAR)

2. Actions

2.1 Present signal tests

   A1: V0 (signal S)
   A2: V1 (signal MOVE_TIME)
```

```
2.2 Output act

   A3: DISPLAY_
   A4: MOVE_CMI

2.3 Assignments

   A5: V4 := (\
   A6: V2 := SC
   A7: V5 := SC
   A8: V6 := ra
   A9: V7 := ra
   A10: V3 := 7

2.4 Conditions

   A11: LESS_TH
   A12: false

3. Automaton

State 0

goto 1

State 1

if A1 then
   A5;
   if A2 then
      A8;A9;A10
      if A11 th
         A6;A3;
         goto 1
      end;
      A4;
      goto 1
   end;
   A7;
   if A11 then
      A6;A3;
      goto 1
   end;
   goto 1
end;
goto 1
```

Multiple proces
interface software.

```
sprintf(cmd,"M


send(cmd);
```

The program m
program then requ

```
2.2 Output actions

   A3: DISPLAY_ALARM (V2)
   A4: MOVE_CMD (V3)

2.3 Assignments

   A5: V4 := (V9 ? V8 : (V9:=true,V8:=S_CURRENT_SONAR())
   A6: V2 := SONAR_TO_R_THETA(V4)
   A7: V5 := SONAR_TO_R_THETA(V4)
   A8: V6 := rand(OMEGA_MIN, OMEGA_MAX)
   A9: V7 := rand(OMEGA_MIN, OMEGA_MAX)
   A10: V3 := TURNS_TO_MOVE(V6, V7)

2.4 Conditions

   A11: LESS_THAN_DISTANCE_TO_DISTANCE(EXTRACT_R(V5), MIN_ALARM_DIST)
   A12: false

3. Automaton

State 0

goto 1

State 1

if A1 then
   A5;
   if A2 then
      A8;A9;A10;A7;
      if A11 then
         A6;A3;A4;
         goto 1
      end;
      A4;
      goto 1
   end;
   A7;
   if A11 then
      A6;A3;
      goto 1
   end;
   goto 1
end;
goto 1
```

Multiple processes can be added to the robot by using the add_process command in the Robuter C interface software. However, a send with APRO works better:

```
sprintf(cmd,"MOVE P RC=%d,%d P=%d \n",move.left_wheel_turns,
                        move.right_wheel_turns,
                                move.period);
send(cmd);
```

The program must be loaded into the robot memory, and the go command issued to start it. The program then requests the user to enter a delay which corresponds to how long the program is to run

(independently monitored). The robot then generates random moves (the number of turns for each wheel is independent) of not more than 20 centimeters a move every ten seconds and stops if an object is detected closer than two centimeters.

References

[1] J. Albus. *Brains, Behavior and Robotics.* BYTE Books, Peterborough, New Hampshire, 1981.

[2] Peter Allen. Integrating Vision and Touch for Object Recognition Tasks. *International Journal of Robotics Research*, 7(6):15–33, 1988.

[3] Russell L. Andersson. Investigating Fast, Intelligent Systems with a Ping-Pong Playing Robot. In Robert C. Bolles and Bernard Roth, editors. *Proceedings of the Fourth International Symposium on Robotics Research*, pages 15–22, MIT Press, Cambridge, Massachusetts, 1988.

[4] Ronald C. Arkin. Motor Schema Based Navigation for a Mobile Robot: An Approach to Programming by Behavior. In *Proceedings of the International Conference on Robotics and Automation*, pages 264–271, IEEE, Raleigh, North Carolina, 1987.

[5] Ronald C. Arkin, Edward M. Riseman, and Allan R. Hanson. AuRA: An Architecture for Vision-Based Robot Navigation. In *Proceedings of the DARPA Image Understanding Workshop*, pages 417–431, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[6] N. Ayache and O.D. Faugeras. Building, Registrating, and Fusing Noisy Visual Maps. *International Journal of Robotics Research*, 7(6):45–65, 1988.

[7] Nicholas Ayache and Olivier D. Faugeras. Building a Consistent 3D Representation of a Mobile Robot Environment by Combining Multiple Stereo Views. In *IJCAI-87*, pages 808–810, Munich, RFA, August 1987.

[8] Nicholas Ayache and Olivier D. Faugeras. Maintaining Representations of the Environment of a Mobile Robot. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 337–350, MIT Press, Cambridge, Massachusetts, 1988.

[9] Gerard Berry and Georges Gonthier. *The Esterel Synchronous Programming Language: Design, Semantics, Implementation.* Research Report 842, INRIA, Sophia Antipolis, France, May 1988.

[10] Bir Bhanu and Wilhelm Burger. DRIVE - Dynamic Reasoning from Integrated Evidence. In *Proceedings of the DARPA Image Understanding Workshop*, pages 581–588, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[11] R. Bhatt, D. Gaw, and A. Meystel. A Real-Time Guidance System for an Autonomous Vehicle. In *Proceedings of the International Conference on Robotics and Automation*, pages 1785–1791, IEEE, Raleigh, North Carolina, 1987.

[12] J.D. Boissonnat, Olivier D. Faugeras, and E. LeBras-Mehlman. *Representing Stereo Data with the Delauney Triangulation.* INRIA Research Report 788, INRIA, Roquencourt, France, February 1988.

[13] Michael Brady, Stephen Cameron, Hugh Durrant-Whyte, Margaret Fleck, David Forsyth, Alison Noble, and Ian Page. Progress toward a System that can Acquire Pallets and Clean Warehouses. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 359–374, MIT Press, Cambridge, Massachusetts, 1988.

[14] Valentino Braintenberg. *Vehicles: Experiments in Synthetic Psychology.* MIT Press, Cambridge, Massachusetts, 1987.

[15] R.A. Brooks. A Robust Layered Control System For A Mobile Robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, March 1986.

[16] Rodney Brooks. A Robust Programming Scheme for a Mobile Robot. In *Languages for Sensor Based Control in Robotics*, pages 509–522, Springer-Verlag, Berlin, 1987.

[17] Robert E. Ca
to Nuclear Fa
on Robotics a

[18] B. Chandrase
mation Proce
1987.

[19] Raja Chatila.
Faugeras and
Research, pag

[20] R. Chattergy.
Research, 4(1)

[21] S.L. Chiu, D.J
of the IEEE
1986.

[22] James L. Cro
pages 793–79

[23] Larry S. Davi
and Phillip V
Image Under
1987.

[24] E.D. Dickmar
Bolles and Be
Research, pag

[25] S.H. Drake. *I*
Massachusett

[26] Bruce A. Dra
Riseman. To
Proceedings o
Inc., Los Alto

[27] Bruno Dufay
on Inductive I
national Sym
1984.

[28] H. F. Durrant
International

[29] H. F. Durran
Robotics Rese

[30] H.F. Durrant-
Proceedings o
CA, April 198

[31] Alberto Elfes.
Automation, 3

[32] Bernard Espia
Applications.
national Sym
1985.

[17] Robert E. Carlton and Stephen J. Bartholet. The Evolution of the Application of Mobile Robotics to Nuclear Facility Operations and Maintenance. In *Proceedings of the International Conference on Robotics and Automation*, pages 720–726, IEEE, Raleigh, North Carolina, 1987.

[18] B. Chandrasekaran. Towards a Functional Architecture for Intelligence Based on Generic Information Processing Tasks. In *Proceedings of IJCAI-87*, pages 1183–1192, Munich, RFA, August 1987.

[19] Raja Chatila. Mobile Robot Navigation: Space Modeling and Decisional Processes. In Olivier D. Faugeras and George Giralt, editors, *Proceedings of the Third International Symposium on Robotics Research*, pages 373–378, MIT Press, Cambridge, Massachusetts, 1986.

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

[21] S.L. Chiu, D.J. Morley, and J.F.Martin. Sensor Data Fusion on a Parallel Processor. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1629–1633, San Francisco, CA, April 1986.

[22] James L. Crowley. Coordination of Action and Perception in a Surveillance Robot. In *IJCAI-87*, pages 793–796, Munich, RFA, August 1987.

[23] Larry S. Davis, Daniel Dementhon, Ramesh Gajulapalli, Todd R. Kushner, Jacqueline LeMoigne, and Phillip Veatch. Vision-Based Navigation: A Status Report. In *Proceedings of the DARPA Image Understanding Workshop*, pages 153–169, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[24] E.D. Dickmann. 4D-Dynamic Scene Analysis with Integral Spatio-temporal Models. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 311–318, MIT Press, Cambridge, Massachusetts, 1988.

[25] S.H. Drake. *Using Compliance in Lieu of Sensory Feedback for Automatic Assembly.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, September 1977.

[26] Bruce A. Draper, Robert T. Collins, John Brolio, Joey Griffith, Allan R. Hanson, and Edward M. Riseman. Tools and Experiments in the Knowledge-Directed Interpretation of Road Scenes. In *Proceedings of the DARPA Image Understanding Workshop*, pages 178–193, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[27] Bruno Dufay and Jean-Claude Latombe. An Approach to Automatic Robot Programming Based on Inductive Learning. In Michael Brady and Richard Paul, editors, *Proceedings of the First International Symposium on Robotics Research*, pages 97–115, MIT Press, Cambridge, Massachusetts, 1984.

[28] H. F. Durrant-Whyte. Consistent Integration and Propagation of Disparate Sensor Observations. *International Journal of Robotics Research*, 6(3):3–24, 1987.

[29] H. F. Durrant-Whyte. Sensor Models and Multisensor Integration. *International Journal of Robotics Research*, 7(6):97–113, 1988.

[30] H.F. Durrant-Whyte. Consistent Integration and Propagation of Disparate Sensor Observations. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1464–1469, San Francisco, CA, April 1986.

[31] Alberto Elfes. Sonar-Based Real-World Mapping and Navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

[32] Bernard Espiau. Closed Loop Control of Robots with Local Environment Sensing: Principles and Applications. In Hideo Hanafusa and Hirochika Inoue, editors, *Proceedings of the Second International Symposium on Robotics Research*, pages 147–154, MIT Press, Cambridge, Massachusetts, 1985.

[33] O.D. Faugeras, N. Ayache, and B. Faverjon. Building Visual Maps by Combining Noisy Stereo Measurements. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1433–1438, San Francisco, CA, April 1986.

[34] Olivier D. Faugeras. Artificial 3D Vision. In *IJCAI-87*, pages 1169–1171, Munich, RFA, August 1987.

[35] B. Faverjon and P. Tournassoud. The Mixed Approach for Motion Planning: Learning Global Strategies from a Local Planner. In *IJCAI-87*, pages 1131–1137, Munich, RFA, August 1987.

[36] Bernard Faverjon and Pierre Tournassoud. Planification et calcul de trajectoires pour Robots Manipulateurs en présence d'obstacles. In *Jounees Geometrique et Robotique*, pages 1–9, INRIA, Toulouse, France, 1988.

[37] William T. Gex and Nancy L. Campbell. Local Free Space Mapping and Path Guidance. In *Proceedings of the International Conference on Robotics and Automation*, pages 424–431, IEEE, Raleigh, North Carolina, 1987.

[38] Malik Ghallab, Rachid Alami, and Raja Chatila. Dealing with Time in Planning and Execution Monitoring. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 431–443, MIT Press, Cambridge, Massachusetts, 1988.

[39] Georges Giralt. Research Trends in Decisional and Multisensory Aspects of Third Generation Robots. In Hideo Hanafusa and Hirochika Inoue, editors, *Proceedings of the Second International Symposium on Robotics Research*, pages 511–520, MIT Press, Cambridge, Massachusetts, 1985.

[40] Georges Giralt, Raja Chatila, and Marc Vaisset. An Integrated Navigation and Motion Control System for Autonomous Multisensory Mobile Robots. In Michael Brady and Richard Paul, editors, *Proceedings of the First International Symposium on Robotics Research*, pages 191–214, MIT Press, Cambridge, Massachusetts, 1984.

[41] M. Goldstein, F.G. Pin, G. de Saussure, and C.R. Weisbin. 3D World Modeling based on Combinatorial Geometry for Autonomous Robot Navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 727–733, IEEE, Raleigh, North Carolina, 1987.

[42] R.A. Grupen. *General Purpose Grasping and Manipulation with Multifingered Robot Hands.* PhD thesis, University of Utah, Merrill Engineering Building, Salt Lake City, UT 84112, August 1988.

[43] Gregory D. Hager. *Active Reduction of Uncertainty in Multisensor Systems.* PhD thesis, University of Pennsylvania, Philadelphia, Pennsylvania, July 1988.

[44] Scott Y. Harmon. The Ground Surveillance Robot (GSR): An Autonomous Vehicle Designed to Transit Unknown Terrain. *IEEE Journal of Robotics and Automation*, 3(3):266–279, June 1987.

[45] T. Henderson and Rod Grupen. Autochthonous Behaviors: Mapping Perception to Action. In Thomas C. Henderson, editor, *NATO ASI on Traditional and Non-Traditional Robotic Sensors*, page to appear, Springer-Verlag, Maratea, Italy, August 28 - September 2 1989.

[46] T.C. Henderson, C.D. Hansen, and Bir Bhanu. The Specification of Distributed Sensing and Control. *Journal of Robotic Systems*, 2(4):387–396, 1985.

[47] T.C. Henderson, Chuck Hansen, and Bir Bhanu. A Framework for Distributed Sensing and Control. In *Proceedings of IJCAI 1985*, pages 1106–1109, Los Angeles, CA, August 1985.

[48] T.C. Henderson and E. Shilcrat. Logical Sensor Systems. *Journal of Robotic Systems*, 1(2):169–193, 1984.

[49] T.C. Henderson, E. Shilcrat, and C.D. Hansen. A Fault Tolerant Sensor Scheme. In *Proceedings of the International Conference on Pattern Recognition*, pages 663–665, August 1984.

[50] Thomas C. Henderson. *Workshop on Multisensor Integration for Manufacturing Automation.* Technical Report UU-CS-87-006, University of Utah, Department of Computer Science, Feb. 1987.

[51] Thomas C. Her
Systems: Inter
1988.

[52] G. Hirzinger. R
and George Gira
pages 155–163,

[53] N. Hogan. Impe
– Applications.
1985.

[54] Can Işik and A
Mobile Robot.

[55] R.A. Jarvis and
Capabilities. In
Symposium on

[56] Leslie P. Kaelbl
Lansky, editors,
Oregon, June 3(

[57] A.C. Kak, B.A.
World Knowled
Conference on

[58] A.C. Kak, A.J.
Int. J. Prod. Re

[59] David J. Kriegn
and Locomotior
pages 402–408,

[60] B. Kuipers and
Summer 1988.

[61] J.-C. Latombe
ometrique et rol

[62] Douglas B. Len
of IJCAI-87, pa

[63] Paul Levi. Princ
ings of the Inte
North Carolina,

[64] Tod S. Levitt, D
In *Proceedings o*
Inc., Los Altos,

[65] John D. Lowran
sensor Integrati

[66] T. Lozano-Pére
Robots. *Interna*

[67] R.C. Luo, M.-H.
Robots. *IEEE J*

[68] Mark B. Metea
cles using Hierar
Robotics and Au

[51] Thomas C. Henderson, Eliot Weitz, Chuck Hansen, and Amar Mitiche. Multisensor Knowledge Systems: Interpreting 3D Structure. *International Journal of Robotics Research*, 7(6):114–137, 1988.

[52] G. Hirzinger. Robot Learning and Teach-In Based on Sensory Feedback. In Olivier D. Faugeras and George Giralt, editors, *Proceedings of the Third International Symposium on Robotics Research*, pages 155–163, MIT Press, Cambridge, Massachusetts, 1986.

[53] N. Hogan. Impedance Control: An Approach to Manipulation: I – Theory, II – Implementation, III – Applications. *ASME Journal of Dynamic Systems, Measurement, and Control*, 107:1–24, March 1985.

[54] Can Işik and Alexander M. Meystel. Pilot Level of a Hierarchical Controller for an Unmanned Mobile Robot. *IEEE Journal of Robotics and Automation*, 4(3):241–255, June 1988.

[55] R.A. Jarvis and J.C. Byrne. An Automated Guided Vehicle with Map Building and Path Finding Capabilities. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 497–504, MIT Press, Cambridge, Massachusetts, 1988.

[56] Leslie P. Kaelbling. An Architecture for Intelligent Reactive Systems. In M.P. Georgedd and A.L. Lansky, editors, *Proceedings of the Workshop on Reasoning about Plans*, pages 395–410, Timberline, Oregon, June 30 - July 2 1986.

[57] A.C. Kak, B.A. Roberts, K.M. Andress, and R.L. Cromwell. Experiments in the Integration of World Knowledge with Sensory Information for Mobile Robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 734–740, IEEE, Raleigh, North Carolina, 1987.

[58] A.C. Kak, A.J. Vayda, R.L. Cromwell, W.Y. Kim, and C.H. Chen. Knowledge-Based Robotics. *Int. J. Prod. Res.*, 26(5):707–734, 1988.

[59] David J. Kriegman, Ernst Triendl, and Thomas O. Binford. A Mobile Robot: Sensing, Planning and Locomotion. In *Proceedings of the International Conference on Robotics and Automation*, pages 402–408, IEEE, Raleigh, North Carolina, 1987.

[60] B. Kuipers and T. Levitt. Navigation and Mapping in Large-Scale Space. *AI Magazine*, 9(2):25–43, Summer 1988.

[61] J.-C. Latombe and C. Laugier. Systèmes de programmation pour la robotique. In *Journees geometrique et robotique*, pages 223–235, INRIA, Toulouse, France, 1988.

[62] Douglas B. Lenat and Edward A. Feigenbaum. On the Thresholds of Knowledge. In *Proceedings of IJCAI-87*, pages 1173–1182, Munich, RFA, August 1987.

[63] Paul Levi. Principles of Planning and control Concepts for Autonomous Mobile Robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 874–881, IEEE, Raleigh, North Carolina, 1987.

[64] Tod S. Levitt, Daryl T. Lawton, David M. Chelberg, and Philip C. Nelson. Qualitative Navigation. In *Proceedings of the DARPA Image Understanding Workshop*, pages 447–465, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[65] John D. Lowrance and Thomas D. Garvey. *Evidential Reasoning: An Implementation for Multisensor Integration.* Technical Note 307, SRI, Inc., December 1983.

[66] T. Lozano-Pérez, M. Mason, and R. Taylor. Automatic Synthesis of Fine-Motion Strategies for Robots. *International Journal of Robotics Research*, 3(1):3–24, 1984.

[67] R.C. Luo, M.-H. Lin, and R.S. Scherp. Dynamic Multisensor Data Fusion System for Intelligent Robots. *IEEE Journal of Robotics and Automation*, 4(4):386–396, August 1988.

[68] Mark B. Metea and Jeffery J.-P. Tsai. Route Planning for Intelligent Autonomous Land Vehicles using Hierarchical Terrain Representation. In *Proceedings of the International Conference on Robotics and Automation*, pages 1947–1952, IEEE, Raleigh, North Carolina, 1987.

[69] A. Meystel. Nested Hierarchical Controller for Intelligent Mobile Autonomous System. In L.O. Hertzberger and F.C.A. Groen, editors, *Intelligent Autonomous Systems*, pages 416–448, North-Holland, Amsterdam, The Netherlands, 1987.

[70] J. Milberg and P. Lutz. Integration of Autonomous Mobile Robots into the Industrial Production Environment. In *Proceedings of the International Conference on Robotics and Automation*, pages 1953–1959, IEEE, Raleigh, North Carolina, 1987.

[71] M. Minsky. *The Society of Mind.* Simon and Schuster, NY, NY, 1986.

[72] A. Mitiche and J.K. Aggarwal. An Overview of Multisensor Systems. *SPIE Optical Computing*, 2:96–98, 1986.

[73] Hans Moravec. Sensor Fusion in Certainty Grids for Mobile Robots. *AI Magazine*, 9(2):61–74, Summer 1988.

[74] Hatem Nasr, Bir Bhanu, and Stephanie Schaffer. Guiding an Autonomous Land Vehicle Using Knowledge-Based Landmark Recognition. In *Proceedings of the DARPA Image Understanding Workshop*, pages 432–439, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[75] James L. Nevins, M. Desai, E. Fogel, B.K. Walker, and D.E. Whitney. Adaptive Control, Learning, and Cost Effective Sensor Systems for Robotic or Advanced Automation Systems. In Michael Brady and Richard Paul, editors, *Proceedings of the First International Symposium on Robotics Research*, pages 983–994, MIT Press, Cambridge, Massachusetts, 1984.

[76] Nils J. Nilsson. *Triangle Tables: A Proposal for a Robot Programming Language.* Technical Report Technical Note 347, SRI International, February 1985.

[77] K. Overton. *The Acquisition, Processing, and Use of Tactile Sensor Data in Robto Control.* PhD thesis, University of Massachusetts, Amherst, Massachusetts, May 1984.

[78] Jocelyne Pertin-Troccaz and Pierre Puget. Dealing with Uncertainty in Robot Planning Using Program Proving Techniques. In Robert C. Bolles and Bernard Roth, editors, *Proceedings of the Fourth International Symposium on Robotics Research*, pages 455–466, MIT Press, Cambridge, Massachusetts, 1988.

[79] M.H. Raibert. *Legged Robots that Balance.* MIT Press, Cambridge, MA, 1986.

[80] Nageswara S.V. Rao, S.S. Iyenfar, C.C. Jorgensen, and C.R. Weisbin. On Terrain Acquisition by a Finite-Sized Mobile Robot in Plane. In *Proceedings of the International Conference on Robotics and Automation*, pages 1314–1319, IEEE, Raleigh, North Carolina, 1987.

[81] K. Salisbury. *Kinematic and Force Analysis of Articulated Hands.* PhD thesis, Stanford, Stanford, California, May 1982.

[82] Steven Salzberg. Heuristics for Inductive Learning. In *IJCAI-85*, pages 603–609, San Francisco, California, August 1985.

[83] R. Schott. On Mobile Robots: A Probabilistic Model for the Representation and Manipulation of Spatial Uncertainty. In *Proceedings of the International Conference on Robotics and Automation*, pages 409–415, IEEE, Raleigh, North Carolina, 1987.

[84] Oliver G. Selfridge and Richard S. Sutton. Training and Tracking in Robotics. In *IJCAI-85*, pages 670–672, San Francisco, California, August 1985.

[85] Grahame B. Smith and Thomas M. Strat. Information Management in a Sensor-Based Autonomous System. In *Proceedings of the DARPA Image Understanding Workshop*, pages 170–177, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[86] R. Smith and P. Cheeseman. On the Representation and Estimation of Spatial Uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1987.

[87] Ralph P. Sobek. A Robot Planning Structure using Production Rules. In *IJCAI-85*, pages 1103–1105, San Francisco, California, August 1985.

[88] Anthony Stent
the DARPA In
California, 198

[89] Charles Thorpe
sion Lab. Visi
Image Underst
1987.

[90] Pierre Tournas
Geometrique et

[91] Ernst Triendl a
ings of the Inte
North Carolina

[92] Ernst Triendl a
Robot. In *Pro*
Kaufmann, Inc.

[93] Saburo Tsuji a
pages 1127–113

[94] F.M. Tuijnmar
Model for Con
Hertzberger an
Holland, Amst

[95] Volker Turau.
Robotics, 4:41–

[96] J.R. White, H.
clear Power Pl
pages 714–719,

[97] William L. Wh
Range-Based M
and Automatio

[98] Dit-Yan Yeung
lem for Multipl
Automation, pa

[88] Anthony Stentz and Yoshimasa Goto. The CMU Navigational Architecture. In *Proceedings of the DARPA Image Understanding Workshop*, pages 440–446, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[89] Charles Thorpe, Steven Shafer, Takeo Kanade, and the members of the Strategic Computing Vision Lab. Vision and Navigation for the Carnegie-Mellon Navlab. In *Proceedings of the DARPA Image Understanding Workshop*, pages 143–152, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[90] Pierre Tournassoud. Motion Planning for a Mobile Robot with a Kinematic Constraint. In *Jounees Geometrique et Robotique*, pages 1–26, INRIA, Toulouse, France, 1988.

[91] Ernst Triendl and David J. Kriegman. Stereo Vision and Navigation within Buildings. In *Proceedings of the International Conference on Robotics and Automation*, pages 1725–1730, IEEE, Raleigh, North Carolina, 1987.

[92] Ernst Triendl and David J. Kriegman. Vision and Visual Exploration for the Stanford Mobile Robot. In *Proceedings of the DARPA Image Understanding Workshop*, pages 407–416, Morgan Kaufmann, Inc., Los Altos, California, 1987.

[93] Saburo Tsuji and Jiang Yu Zheng. Visual Path Planning by a Mobile Robot. In *IJCAI-87*, pages 1127–1130, Munich, RFA, August 1987.

[94] F.M. Tuijnman, W. Beemster, W. Duinker, L.O. Hertzberger, E. Kuipers, and H. Muller. A Model for Control Software and Sensor Algorithms for an Autonomous Mobile Robot. In L.O. Hertzberger and F.C.A. Groen, editors, *Intelligent Autonomous Systems*, pages 610–615, North-Holland, Amsterdam, The Netherlands, 1987.

[95] Volker Turau. A Model for a Control and Monitoring System for an Autonomous Mobile Robot. *Robotics*, 4:41–47, 1988.

[96] J.R. White, H.W. Harvey, and K.A. Farnstrom. Testing of Mobile Surveillance Robot at a Nuclear Power Plant. In *Proceedings of the International Conference on Robotics and Automation*, pages 714–719, IEEE, Raleigh, North Carolina, 1987.

[97] William L. Whittaker, George Turkiyyah, and Martial Hebert. An Architecture and Two Cases in Range-Based Modeling and Planning. In *Proceedings of the International Conference on Robotics and Automation*, pages 1991–1997, IEEE, Raleigh, North Carolina, 1987.

[98] Dit-Yan Yeung and George A. Bekey. A Decentralized Approach to the Motion Planning Problem for Multiple Mobile Robots. In *Proceedings of the International Conference on Robotics and Automation*, pages 1779–1784, IEEE, Raleigh, North Carolina, 1987.