Logical Behaviors[1]

Thomas C. Henderson

UUCS-89-012

Department of Computer Science
University of Utah
Salt Lake City, UT 84112 USA

December 27, 1989

Abstract

In this paper we describe an approach to high-level multisensor integration in the context of an autonomous mobile robot. Previous papers have described the development of the INRIA mobile robot subsystems:

1. **sensor and actuator systems**

2. **distance and range analysis**

3. **feature extraction and segmentation**

4. **motion detection**

5. **uncertainty management**, and

6. **3-D environment descriptions.**

We describe here an approach to:

- the **semantic analysis** of the 3-D environment descriptions.

Knowledge Base

Goals & Task Compilation

World Model
* Self
* Topological
* Metrological

Procedural Behaviors

Sensors,
Actuators,
Control
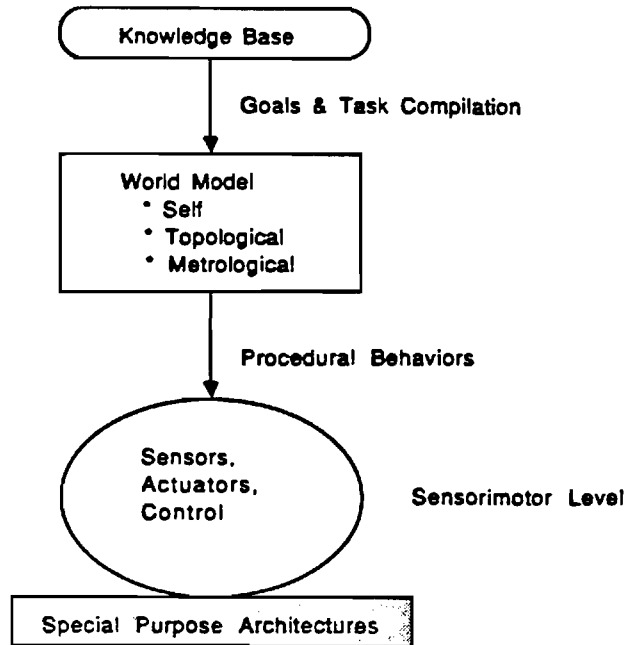
Sensorimotor Level

Special Purpose Architectures

Figure 1: Autonomous Robot Vehicle Research

1 Introduction

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 the second problem.

Autonomous mobile robots have been studied in a wide range of contexts. Figure 1 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 [16,18,33,36,44,48,58,69,70,74,77]. Most such systems must use sensors (e.g., sonar or cameras [21,24,27,76]) and actuators and must control them [25,43,57,62,79,80]. The use of sensors requires the study of uncertainty management [35, 64,68,71] and multisensor integration [17,23,37,54,55,59]. More global approaches to the sensorimotor problem can be found in [1,9,32], and special purpose architectures are being planned [3,73,82].

One level up, the mapping of procedural behaviors onto the sensorimotor control structure is of interest [2,4,12,39,38,41,42,78]. The world representations also exist at this level:

both the metrological [5,6,60,65], where precise measurement is paramount, and topological [8,10,15,28,29,30,31,34,45,52,56,72,75,83], where adjacency relations are useful for path planning, etc. It is even possible to study primitive forms of learning in this context [67,75].

Broader studies are usually oriented towards particular applications (e.g., the nuclear industry [13,81], road following [19,20,61]) or towards well-defined, but limited goals (e.g., indoor [11,26] or outdoor [49,53] navigation).

Finally, the 'highest' level involves the specification and representation of the knowledge appropriate to a given task [14,46,47,51,63] and its compilation into executable robot behavior (or programs) [22,40,50]. 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 balckboard 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.

2   Problem Definition

We 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).

In this paper 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 "intelligent" 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.).
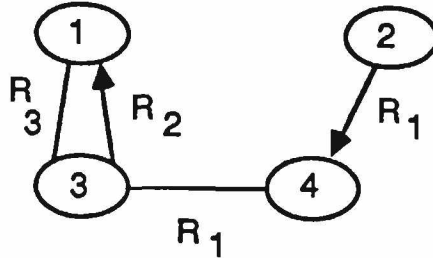
Figure 2: Semantic Net Defining World Model

3   Logical Behaviors

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 2). 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 3). 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. Techniques for the satisfaction of the relations are called **logical behaviors**.

As an example, consider the world model in Figure 4 which represents a specific office at INRIA. 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 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.

3.1   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.
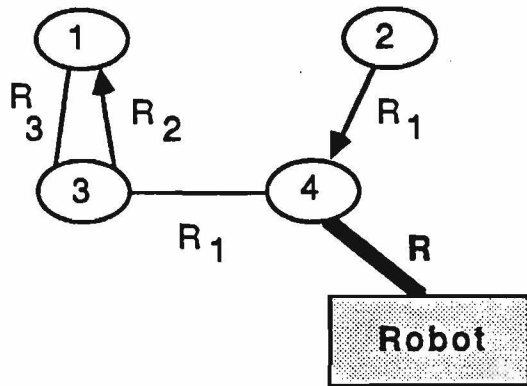
4

Figure 3: Defining the Robot Task

Next_to: ....
Parallel: ▬▬▬▬
Perpendicular: ────
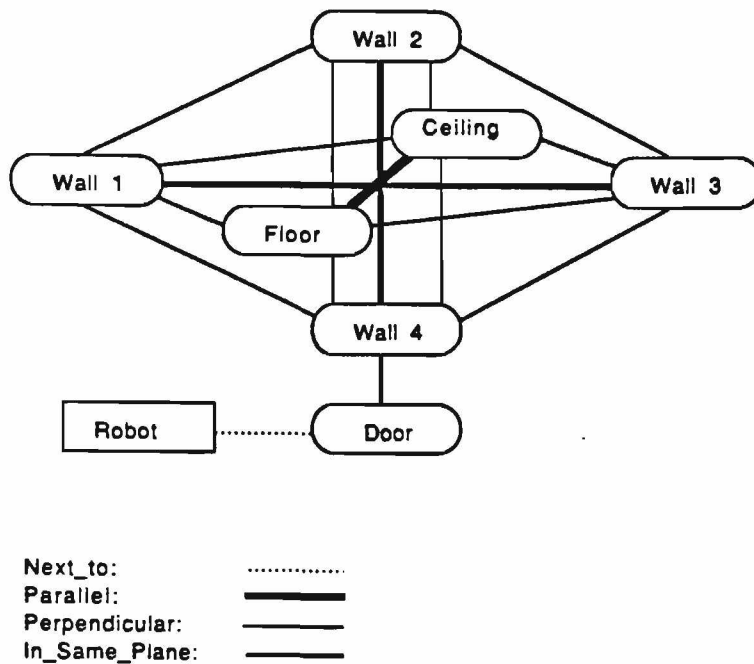In_Same_Plane: ────

Figure 4: A Representation of the Command: "Go to the office door"

We have selected the Esterel synchronous programming language[7] 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 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**).

6

Prior Knowledge	Robot World	Sensors		External Windows

* Source
 	
  - POP11 Code
  - Lisp Code
  - Prolog Code
 	
* CAD Tool
 	
* Other
	* Behavior
	
	  - Automaton
	  - Trace
	  - Robot Dump
	
	* Goals, State
	
	* Maps, Objects
		* Sonar
		
		  - Range
		  - Direction
		
		* Motors
		
		* 3D Segments
				* Camera Images
				
				* Edge Images
				
				* Etc.

POPLOG				Suntools

Figure 5: The Debugging System Organization

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.

3.2 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 5 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 6 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)

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

```
CONSOLE
bon jour,
...more...
                                        /bin/tcs
PCFLOG Arnode Manager   version 1.78  pcp11
: ved Room_C838.pl


: sonar_win_man();

---< 81> ved Room_C838.pl -----------------------------------
    A1: V8 (signal S)
    A2: V1 (signal MOVE_TIME)

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)
    A18: 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 8

goto 1

State 1

if A1 then
    A5;
    if A2 then
        A8;A9;A18;A7;
        if A11 then
            A6;A3;A4;
            goto 1
        end;
        A4;
        goto 1
    end;
    A7;
```

```
~u/corse/1/ptch/if/osolog/develop/Room_C838.pl
---< 45> ved Room_C838.pl -----------------------------------

wall(X) :-   plane(X),
             /* perpendicular_planes(X,Ceiling), ceiling(Ceiling), */
             /* perpendicular_planes(X,Floor), floor(Floor), */
             /* block_passage(X), */
             vertical_plane(X).

door(X) :-   plane(X),
             /* permits_passage(X), */
             in_plane(X,Wall), wall(Wall).

ceiling(X) :-   plane(X),
             /* blocks_passage_above(X), */
             /* perpendicular_planes(X,Wall), wall(Wall), */
             /* parallel_planes(X,Floor), floor(Floor), */
             highest_horizontal_plane(X).

plane(X) :- plane_obj(X,XX,XY,XZ,XD).
```

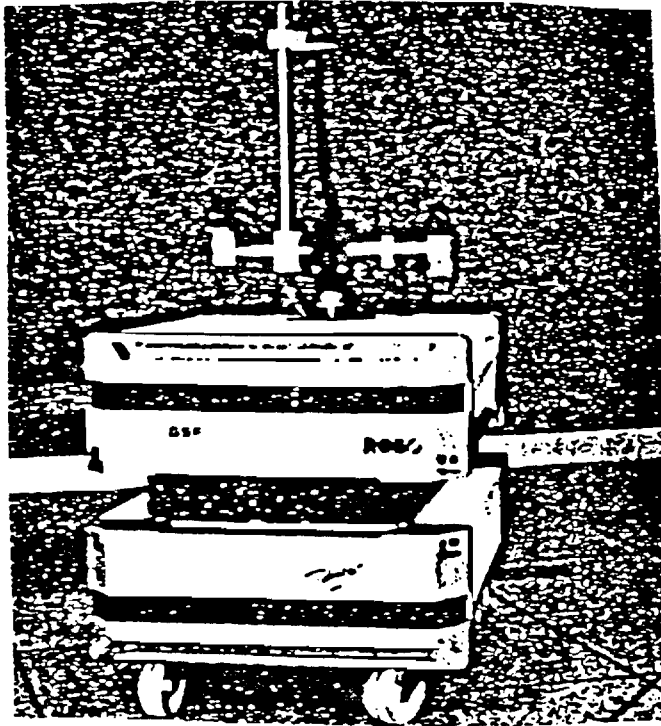Figure 6: Collection of Windows for Debugging

Figure 7: The INRIA Mobile Robot

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.

4 Implementation

4.1 Mobile Robot

Figure 7 shows the operational mobile robot at INRIA. It is similar to other mobile robots (e.g., like those at CMU or Hilare at LAAS). Figure 8 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.

A graphical interface has been developed which permits a model of the ground floor to be specified and for the robot to be instructed to move in that envirnoment while avoiding obstacles. Figure 9 shows a session at the Rocquencourt location of INRIA. For full details, see [66].

4.2 Building Environment Descriptions

Many papers have been published describing our methods for building robust environment descriptions[5,6,26,27]. Current capabilities include 3-camera stereo and robust multi-view fusion.
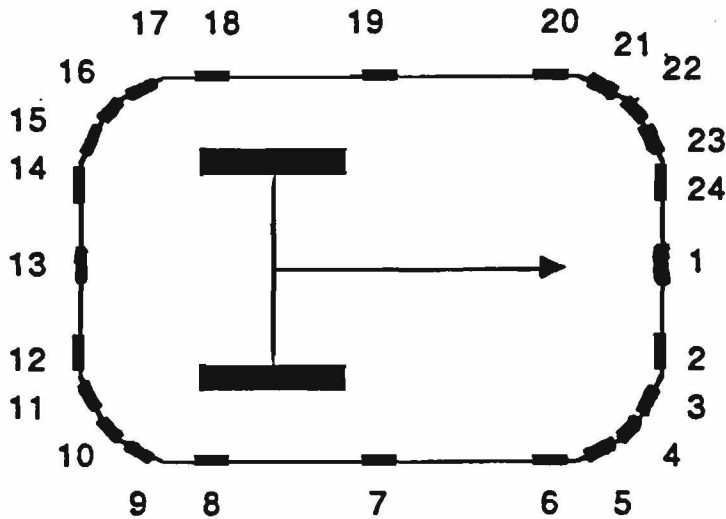
Figure 8: The Geometry and Sensor Placement on the INRIA Mobile Robot

Figure 10 shows a typical office scene and Figure 11 shows a set of 3-D segments reconstructed from the analysis of such a scene. This 3-D description provides the basis for the development of logical sensors for object recognition and localization.

5 Summary and Future Work

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.

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
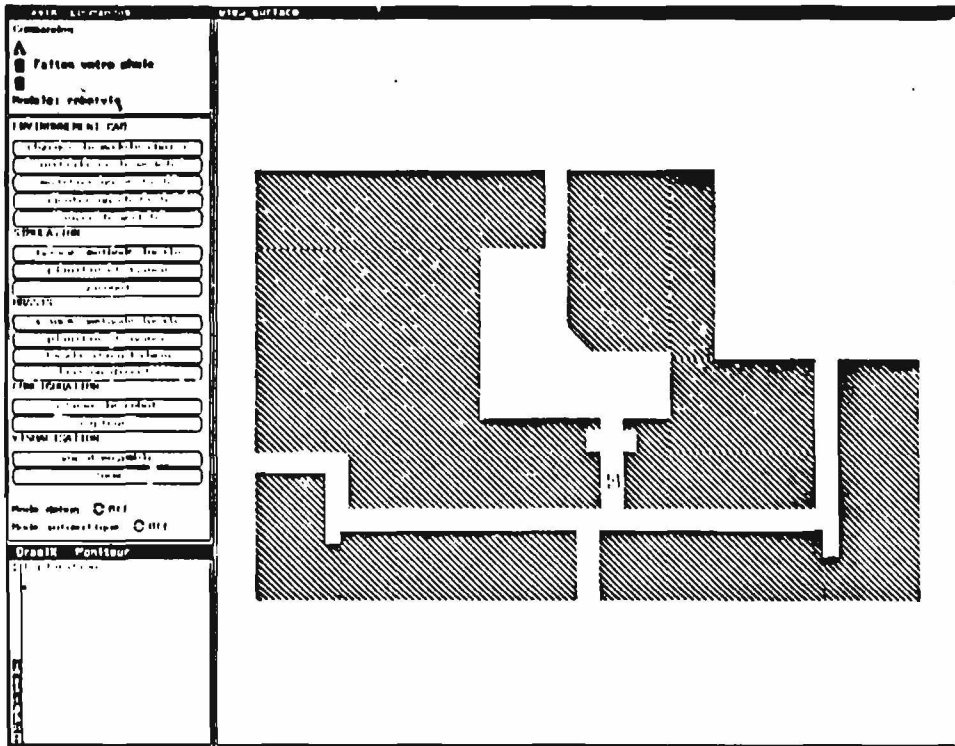
Figure 9: Graphical Interface to the Mobile Robot

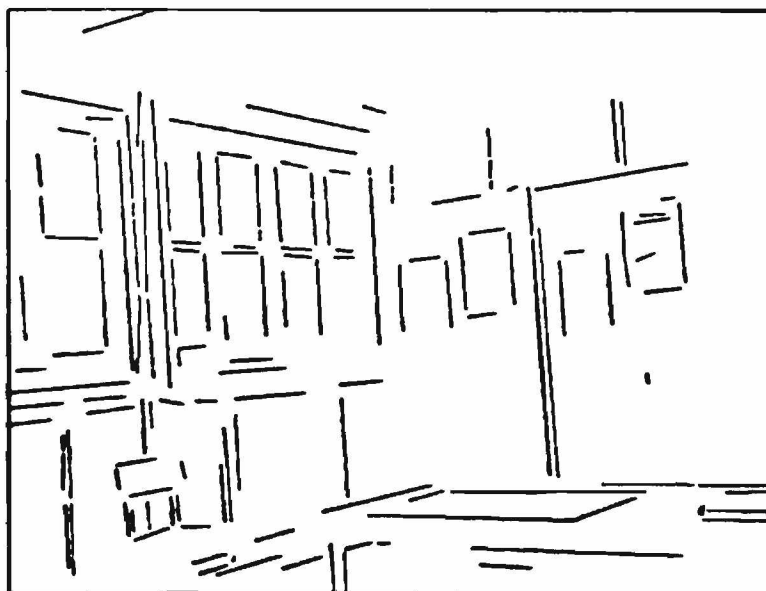Figure 10: Typical Office Scene

Figure 11: 3-D Segments Recovered from Scene

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)

15

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 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] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

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

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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, California, April 1986.

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

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] Hugh 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, California, April 1986.

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

[25] 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.

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

[27] Olivier D. Faugeras, Nicholas Ayache, and Bernard Faverjon. Building Visual Maps by Combining Noisy Stereo Measurements. In *Proceedings of the IEEE Conference on Robotics and Automation*, pages 1433–1438, San Francisco, California, April 1986.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

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

[36] 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.

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

[38] Thomas C. Henderson, Chuck D. Hansen, and Bir Bhanu. A Framework for Distributed Sensing and Control. In *IJCAI-85*, pages 1106–1109, San Francisco, California, August 1985.

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

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

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

[42] Thomas C. Henderson, Esther Shilcrat, and Chuck D. Hansen. A Fault Tolerant Sensor Scheme. In *Proceedings of the International Conference on Pattern Recognition*, pages 663–665, Montreal, August 1984.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

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

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

[52] 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.

[53] 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.

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

[55] 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.

[56] 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.

[57] 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.

[58] 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.

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

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

[61] 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.

[62] 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.

[63] Nils Nilsson. *Triangle Tables: A Proposal for a Robot Programming Language.* Technical Note 347, Stanford University, December 1985.

[64] 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.

[65] 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.

[66] J.L. Robles. *Planification de trajectoires et évitement d'obstacles pour un robot mobile équipé de capteurs à ultrasons.* DEA d'Informatique , Université de Paris Sud, September 1988.

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

[68] 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.

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

[70] 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.

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

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

[73] 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.

[74] 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.

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

[76] 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.

[77] 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.

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

[79] 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.

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

[81] 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.

[82] 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.

[83] 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.