

UNCLASSIFIED

AD NUMBER: AD0841509

LIMITATION CHANGES

TO:

Approved for public release; distribution is unlimited.

FROM:

This document is subject to special export controls; 1 Jun 1968, and each transmittal to foreign governments or foreign nationals or their representatives may be made only with prior approval of RADC (EMIRC), Griffiss AFB, NY, 13440.

AUTHORITY

ST-A RADC, AFSC LTR, 14 OCT 1971

AD841509

RADC-TR- 68-196



APPLICATION OF INTELLIGENT AUTOMATA TO RECONNAISSANCE

N. Nilsson  
B. Raphael  
S. Wahlstrom

Stanford Research Institute

TECHNICAL REPORT NO. RADC-TR- 68-196  
June 1968

This document is subject to special  
export controls and each transmittal  
to foreign governments, foreign na-  
tionals or representatives thereto may  
be made only with prior approval of  
RADC (EMIRD), GAFB, N.Y.

Rome Air Development Center  
Air Force Systems Command  
Griffiss Air Force Base, New York



When US Government drawings, specifications, or other data are used for any purpose other than a definitely related government procurement operation, the government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded, by implication or otherwise, as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacturer, use, or sell any patented invention that may in any way be related thereto.

ACCESSION for	
CPSTE	WHITE SECTION <input type="checkbox"/>
DDC	BUFF SECTION <input checked="" type="checkbox"/>
UNANNOUNCED	<input type="checkbox"/>
JUSTIFICATION	
BY	
DISTRIBUTION/AVAILABILITY CODES	
DIST.	AVAIL. and/or SPECIAL
2	

Do not return this copy. Retain or destroy.

**APPLICATION OF INTELLIGENT AUTOMATA TO RECONNAISSANCE**

**N. Nilsson  
B. Raphael  
S. Wahlstrom**

**Stanford Research Institute**

**This document is subject to special  
export controls and each transmittal  
to foreign governments, foreign na-  
tionals or representatives thereto may  
be made only with prior approval of  
RADC (EMIRD), GAFB, N.Y. 13440.**

## FOREWORD

This report was submitted by Stanford Research Institute, Menlo Park, California under Contract AF30(602)-4147, Project 4594, Task 459405. It covers research performed by the contractor during the period 18 December 1967 - 5 April 1968. The number assigned to the report by the contractor for internal use was SRI Project 5953.

Professional personnel associated with the various tasks on this project during the past five months were as follows:

Task 1, Visual System; G. Forsen

H. Chan, R. Loda, P. Hart, B. Jackson

Task 3, Programming; Leader: B. Raphael

L. Chaitin, L. Coles, J. Harris, B. Jackson, J. Wensley

Task 5, Special Equipment; Leader: S. Wahlstrom

J. Baer, P. Johanson, J. Yarborough

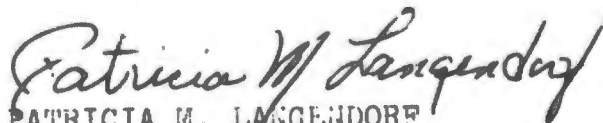
Task 7, Program Management; Leader: C. Rosen

N. Nilsson

The project engineer was Patricia M. Langendorf, Rome Air Development Center, EIRDC, Griffiss Air Force Base, New York 13440.

This technical report has been reviewed and is approved.

Approved:



PATRICIA M. LANGENDORF  
Project Engineer

Approved:



WILLIAM B. MOORE  
Chief, Recon Intel Data Handling Br

## ABSTRACT

---

This report describes the results of research during the past four months on the project, "Applications of Intelligent Automata to Reconnaissance." The primary goal of this project is to investigate techniques in artificial intelligence applied to the control of a mobile automaton in a real environment. The main emphasis is on the design of a hierarchy of computer programs that will accept visual and other sensory information gathered by the automaton and will direct the actions of the automaton in performing missions that require the abilities to plan ahead and to learn from previous experience.

The work described has largely been aimed toward preparing for an extensive series of experiments with the automaton system, and completes all of the items specified under Phase II of the project. This report summarizes our present status in software and hardware development and then outlines the purposes and plan for the experiments to follow.

## CONTENTS

---

I	INTRODUCTION . . . . .	1
II	SOFTWARE DEVELOPMENT . . . . .	4
	A. Present Status. . . . .	4
	B. Vehicle Control . . . . .	4
	C. Modeling. . . . .	4
	D. Navigation. . . . .	5
	E. Monitoring and Software Aids. . . . .	6
	F. Vision. . . . .	6
	G. Control Language. . . . .	7
III	HAE WARE DEVELOPMENT . . . . .	9
	A. TV Preprocessor . . . . .	9
	B. Additional Controls for TV Camera . . . . .	9
	C. The Vehicle . . . . .	9
	D. Communication . . . . .	10
IV	PLANS FOR EXPERIMENTS. . . . .	11
	A. Purpose of the Experiments. . . . .	11
	B. The Plan. . . . .	14
	1. Phase A. . . . .	14
	2. Phase B. . . . .	15
	3. Phase C. . . . .	16
	Appendix--DATA-COMMUNICATION-INTERFACE LOGIC. . . . .	18
	REFERENCES. . . . .	24

## ILLUSTRATIONS

---

Fig. 1	Automaton Vehicle. . . . .	2
Fig. A-1	Information-Exchange Paths . . . . .	20
Fig. A-2	Data-Communication Interface (Fixed) . . . . .	21

## I INTRODUCTION

The primary goal of this project is to investigate techniques in artificial intelligence applied to the control of a mobile automaton in a realistic environment. The main emphasis is on the design of a hierarchy of computer programs that will accept visual and other sensory information gathered by the automaton and will direct the actions of the automaton in performing missions that require the abilities to plan ahead and to learn from previous experience.

The project began in March 1966 and, since that time, three interim reports have been written<sup>1,2,3\*</sup> as well as two short papers<sup>4,5</sup> that present overviews of the project. In this report we shall discuss the work performed since December 1967 (referring the reader to the relevant sections of the previous reports for background). We shall also describe a plan for experiments to be conducted during the remainder of the project.

We are now experimenting with a mobile vehicle that is controlled by an SDS-940 computer over a cable link (see Fig. 1). This vehicle is discussed in detail in the First Interim Report<sup>1</sup> (pp. 56-67), the Second Interim Report<sup>2</sup> (pp. 27-31), and in Appendix A of the Third Interim Report.<sup>3</sup> The SDS-940 computer complex is described in the First Interim Report (pp. 11-15). In addition, other special hardware has been constructed, principally a TV preprocessor that is described in the First Interim Report (pp. 68-91 and pp. J1 through J-7) as well as in a paper by Forsen.<sup>6</sup> Also we are presently installing a radio channel between the vehicle and the computer that will replace the cable link. The present automaton software organization is thoroughly described in the Third Interim Report.

---

\*References are listed at the end of this report.

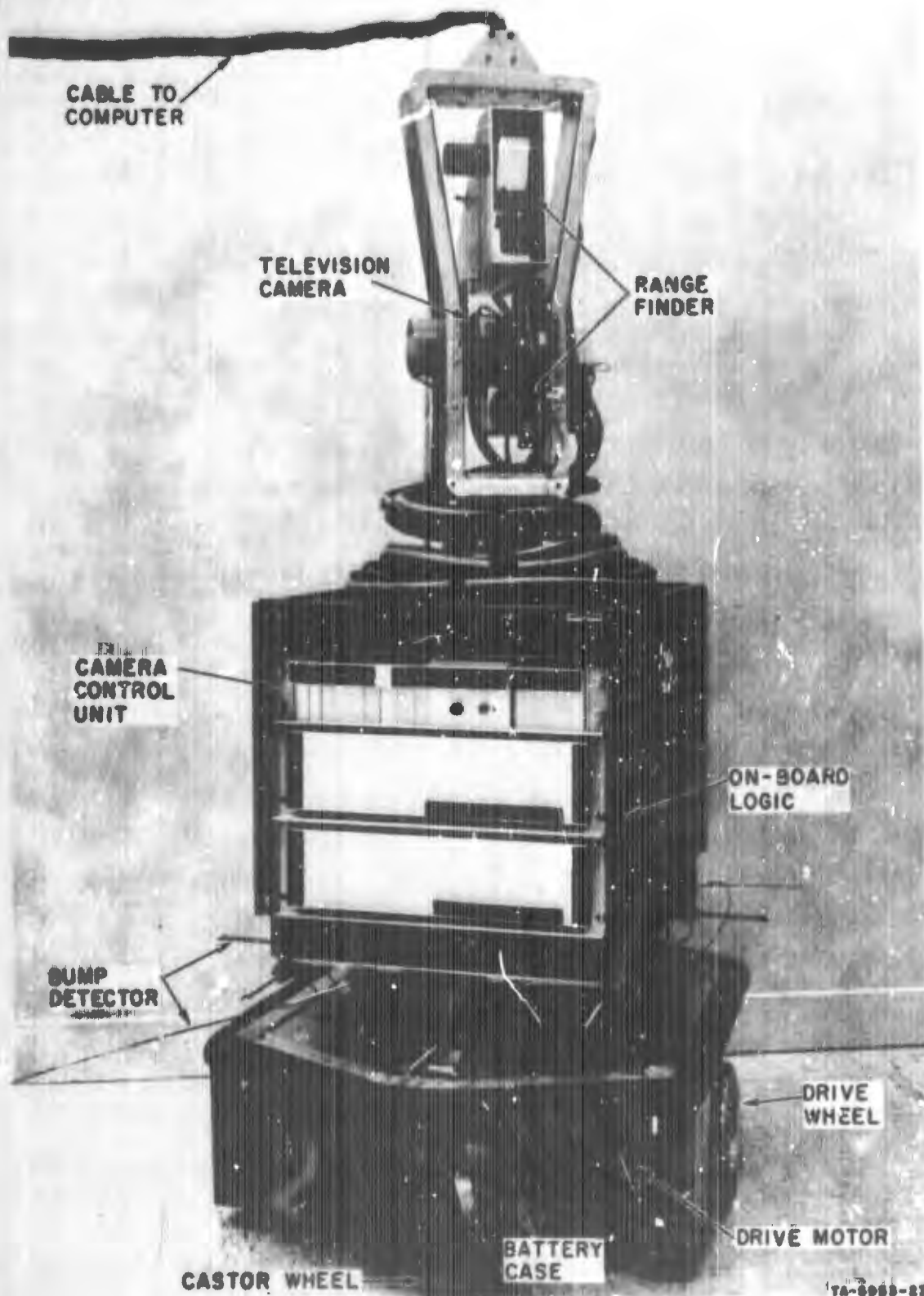


FIG. 1 AUTOMATON VEHICLE

On 1 November 1967, the scope of the project was expanded to allow more thorough experimentation with a system having enhanced problem-solving and visual capabilities. Our efforts during the past four months have been largely in preparation for these experiments and can be grouped under three headings:

- (1) Software development.
- (2) Hardware development.
- (3) Preparation of a plan for the experiments to follow.

Accordingly, the following sections of this report will deal with these three topics.

## II SOFTWARE DEVELOPMENT

### A. Present Status

The automaton software system currently requires about 40,000 words of computer memory. The programs that we will briefly review in this section are conveniently separable into different categories, and fortunately (through the operation of our "valet" system) they do not all have to be in core simultaneously. All of the programs to be described here are operational as of 30 April 1968.

In parentheses after each major heading are references to that section of the Third Interim Report where more details can be found.

### B. Vehicle Control (see also pp. 25-28, Third Interim Report)

The automaton vehicle is completely under computer control. Commands can be issued to control power to vehicle modules, move either or both wheels, pan and tilt the camera and range-finder assembly, operate the range-finder, set the camera's iris and focus, and specify whether a cat-whisker bump should cause a reflex action that stops the wheels. Various conditions including bumps and completion of actions cause interrupt signals to be sent to the computer, and various registers on the vehicle including wheel counters and bump status may be read by the computer.

These vehicle-control programs may all be called and controlled from FORTRAN programs and may all be operated under the time-sharing monitor system on the SDS-940 computer without causing any unusual load on the system. In addition, several vehicle actions may be executed simultaneously and may be "overlapped" with computation by the main program.

### C. Modeling (see also pp. 9-14, Third Interim Report)

The automaton's model of its environment presently consists of a hierarchical grid of cells representing regions of a floor plan. Each region is identified as being known occupied, known empty, unknown,

subdivided, or partially occupied (and not subdivided). Each level of the hierarchy contains 16 cells; the maximum depth of subdivisions is a program parameter.

The model is generally initialized with a top-level grid of 16 unsubdivided cells, each marked as unknown. Then, as the vehicle moves about or takes range-finder sightings, the knowledge gained about regions traversed (and therefore known empty areas) and obstacles encountered (and therefore partially occupied areas) is added to the model. Existing programs make it possible to update the model in a variety of ways. Future experimentation will be necessary to determine how best to use new information obtained by the sensors during exploratory journeys.

D. Navigation (see also pp. 20-31, Third Interim Report)

Two programs are available to move the automaton to any specified (x,y) position: One uses a "blind" edge-following heuristic, and the other uses the model to plan a route before starting.

The "blind" navigation program uses only the bump sensors and dead-reckoning knowledge, based on information from the wheel-position registers, of where the vehicle is at any time. This program causes the vehicle to move in a straight line toward its goal until it bumps into something, and then to try to move around the obstacle until it is clear. Although such a program uses very little computation, it has no memory and therefore can get trapped in cul-de-sacs and repeatedly blunder into the same obstacles.

The journey-planning and execution routines are much more sophisticated. They make use of whatever information is available in the model. The planning routine generates a path of straight line segments that the automaton could traverse to the goal without encountering any known obstacles. This path is a minimum-cost path, where "cost" is measured by the sum of distance through known-empty regions plus  $k$  times distance through unknown regions;  $k$  is a parameter of the program. Note that if  $k$  is large, the automaton will prefer to traverse a longer "safe" route, whereas if  $k$  is small the automaton will prefer to explore unknown regions.

The execution routine attempts to carry out a plan, and reports back to its calling program if any difficulties, such as new obstacles, are encountered. Future experimentation is needed to determine the tradeoffs between (1) mapping the environment with extensive explorations and range-finder scans, and (2) early goal-directed tasks with information-gathering as an incidental by-product.

#### E. Monitoring and Software Aids

An on-line CRT display of the model has been useful in several aspects of our program development. We can now easily observe the automaton's view of the world, and compare it with the actual environment. In addition, the display can show the positions of the vehicle and its goal, and the path it has just traversed or plans to traverse.

Our present programs already require more than three times as much computer memory space than is directly available to any one user of the SDS-940 system. Therefore, our "valet" system, which allows complete sets of programs to call upon each other as if they were subroutines, has been essential to our progress. This system has been used primarily for communication among sets of FORTRAN programs. However, we are now extending it to permit communication between FORTRAN and LISP routines.

The high-level control-language and problem-solving aspects of the automaton software (to be discussed below) will be programmed in LISP. Considerable developmental research and experimentation is still needed to determine how those programs will interface with the FORTRAN navigation and modeling programs. In particular, the detailed structure of a LISP version of the model of the environment, and precisely how much information will be transferred between the LISP and FORTRAN models, remains to be determined. The LISP "valet" facility is a vital first step in this direction.

#### F. Vision (see also pp. 15-19, Third Interim Report)

Our analysis and use of information from the TV camera is largely dependent upon the preprocessor hardware, which is now being checked out. Therefore we have temporarily concentrated on programs that

bypass the preprocessor by reading picture data directly into main computer memory, and then simulating the action of the preprocessor in differentiating and masking the data. These programs, which also can display results of various stages of preprocessing on the CRT, are now operational. They are useful for helping us check out the preprocessor hardware as well as for permitting us to proceed with higher-level software before the basic hardware is ready.

An obvious first step toward using the mask responses for scene analysis is to construct a line drawing of the scene. Programs now exist to connect approximately collinear masks into groups and to fit a straight line to each group. Much experimentation will be necessary to develop techniques for constructing a complete line drawing out of these long line segments.

Visual information will initially be transferred to the grid model through use of a program called "floor boundary." This program, just being completed, uses the long line segments extracted from the picture to construct a rough outline of foreground area on the ground plane. The fact that this area was "seen" to be unoccupied is then used to mark appropriate grid cells as known empty.

In parallel with this picture-processing effort, some work has been done recently on simple heuristics for recognizing objects directly from the raw (nonpreprocessed) picture data. For example, when the scene contains only one object a program that merely recognizes and counts vertical lines can distinguish cubes from other shapes of objects. To determine the generality and practicality of such an approach we shall require further study and experimentation.

#### G. Control Language

Eventually the automaton should be able to solve problems stated in a high-level control language, possibly resembling natural English. The linguistic and problem-solving capabilities needed are so different from the mapping and navigational capabilities described above that

they have been developed until now as independent studies. This work applies some related research supported by AFCRL under Contract AF 19 (628)-5919 (SRI Project 6001).

The first-order predicate calculus has been chosen as our internal language for representing desired states and actions of the automaton. This formal, unambiguous language is well-defined and well-understood by mathematicians. It represents a convenient target language for translation from an English-like user-oriented problem-statement language, and a convenient source language for the problem-solving programs that must interface to the established modeling and navigation system.

The English-to-predicate calculus translation will be performed by the method of syntax-directed interpretation. The basic program for carrying this out is now operational. It presently recognizes the syntax of a simple phrase-structure subset of English, and contains the semantic rules for statements about set-inclusion and set-membership relations. The principal addition required to make this program applicable to the automaton is to extend its semantics to include spatial relationships relevant to the automaton's environment.

Two approaches are being pursued in relating predicate calculus statements to particular automaton situations. The first involves using a formal theorem-proving program as the central problem-solving mechanism. A general proof procedure has been programmed, and methods of applying it to problem solving and plan-generation tasks are in the final developmental stages. However, preliminary experiments have demonstrated the impracticality of this approach until we complete current work toward improving the efficiency of these techniques.

The second approach, just now getting started, involves the development of an ad hoc "interpreter" program that can relate a predicate statement to the specific actions of which the automaton is capable. Considerable experimentation will be necessary to determine the most appropriate ways in which the high-level problem-solving and the low-level navigation and model-building programs may be linked.

### III HARDWARE DEVELOPMENT

The work on special hardware during the last few months has been concentrated on the four items discussed below.

#### A. TV Preprocessor

The checkout of the TV preprocessor has reached a stage where all operations can be initiated from the computer. Data transfers between computer and the core memory in the TV preprocessor have been tested with varying test patterns for data and location, with accurate results. All the internal sequences in the preprocessor except for data acquisition from the video input have been checked. Certain of these sequences are dependent on the sequence of data transferred from scratch-pad memory and the core memory. The checking of this data dependence has started but is not completed. The checkout of the transfer of data from core memory to scratch-pad memory required extra time, due to a marginal power supply and incompatible logic levels. The work on the data-acquisition part of the preprocessor has just started, with design of the circuits for separation of horizontal and vertical synchronizing signals from the receiver composite video signal.

#### B. Additional Controls for TV Camera

The additional controls for the TV camera are arranged to set target voltage, beam current, and gain of the video signal on command from the computer. A special circuit for measuring excursions of the video signal above and below fixed thresholds is also included. The circuits have been wired but not yet checked out.

#### C. The Vehicle

The coupled-mode operation for the wheels on the vehicle has been subject to a sequence of modifications to overcome a problem of instability. The problem was finally solved by replacing the electronic coupling with a mechanical link between the two driving wheels. Two electromagnetic clutches are used, one for coupling when the

wheels are rotating in the same direction, and one for coupling when the wheels are rotating in the opposite direction. In noncoupled mode, both clutches are disengaged.

#### D. Communication

We are in the process of replacing the cable linking the automaton vehicle and the computer. This radio link involves two parts--one for transmitting low-bandwidth digital data between the vehicle and computer, and one for transmitting a video image from the TV camera on the vehicle to the computer. The channel for digital data and its associated logic is discussed in the appendix to this report.

The vehicle signal from the TV camera on the vehicle will be transmitted with a wideband FM transmitter at a frequency of approximately 1700 to 1800 MHz. The transmitter and the receiver have been ordered from RHC Electronics Laboratory Inc. The transmitter has a physical dimension that is compatible with our requirements on the vehicle. The receiver has been ordered from the same manufacturer to guarantee compatibility in the link. Circuits for sync-separation have been designed and built. These circuits will connect to the data-acquisition part of the TV preprocessor.

## IV PLANS FOR EXPERIMENTS

### A. Purpose of the Experiments

It is anticipated that the final several months of this project will be devoted mainly to conducting experiments with the automaton vehicle using the extensive array of software and techniques already developed. There are two major reasons for conducting these experiments. First, and most important, it is necessary to gain experience with automaton systems. Since no such system having the same complexity exists anywhere, no one yet has the experience necessary to make informed decisions about the course of future research in this area. Second, our research so far has posed several explicit questions that can best be answered experimentally. Some of these are:

- (1) What programming languages ought to be used for various parts of the system? In particular, how much of the system needs to be written in machine code; when can we use a higher-level language like FORTRAN; and when are list-processing languages such as LISP advantageous? What should happen at the interfaces between parts of the system programmed in different languages? How should control and data be transferred from a program in one language to a program in another? Is the price extracted by these interface problems too much to pay for the advantages of letting each member of the research team use that language he feels is most appropriate to his part of the system?
- (2) What role should special-purpose hardware (such as the TV preprocessor) play? Can special "programming tricks" obviate the need for such special hardware? Can dependence on vision, for example, be reduced by, say, improved memory so that the computations required by vision don't have to occur so frequently? Alternatively, does the

availability of special visual hardware allow vision to be used often enough so that valuable memory can be freed for other important purposes?

- (3) How should information gathered about the world be stored in a model or models? What types of data structures are useful for these models? If more than one model is used, how should information be transferred among the models?
- (4) How should the model of the world be updated by new information obtained by the sensors? Should every piece of memory information be checked against the model to see if the system could have predicted it? If so, should the model always be updated when prediction would have failed? What about special model utility routines that perform background processing on the model to rearrange the information it contains? Should model updating be regarded as a secondary activity taking advantage of new information that happens to come in while the automaton is attempting to perform some primary task, or should explicit information-gathering journeys be attempted?
- (5) How thoroughly and to what detail should the automaton attempt to calculate a route before actually beginning the first part of its journey? To what extent should the automaton avoid or prefer "unknown" areas in calculating journeys? Can this preference be effectively regulated by a single parameter? If so, how should this parameter depend on such factors as the importance of the journey, the present level of knowledge of the model, etc.
- (6) Will the visual technique of converting a scene into a line drawing and then into a floor boundary work effectively in the environment planned for the automaton? How complex an environment can the current visual system handle? How should objects in the visual field be recognized? Will simple pattern-recognition techniques

work well enough to be useful? If not, how should they be elaborated? How should we deal with wider ranges of illumination levels and the possibility of one object occluding another?

(7) What types of problem-solving methods will work best? Will the route-finding algorithm presently programmed be adequate? Would a simpler (but less powerful) one suffice? What about more complex problems involving going to implicitly specified destinations or moving objects about? To what extent will powerful problem-solving methods based on an axiomatization of the problem domain and formal theorem-proving techniques be useful? Might they be capable of being used generally for all problem-solving activities in the automaton system? If so, can the entire automaton executive system be organized as a formal problem-solver? Otherwise, for what problems will ad hoc problem-solving methods be necessary?

(8) How should the automaton executive system be organized if it turns out to be inappropriate to base it on formal problem-solving methods? How should it communicate with the experimenter and with the various problem-solving, model-maintaining, and sensory-processing routines? Is the organization we are presently using capable of growth (that is, can extensive modifications to the problem-solving and other routines be made without necessitating a radical change in the executive system)?

Obviously definitive answers to questions like these cannot be obtained in just a few months of experimenting, but we do hope to make progress on understanding the issues behind these questions and at obtaining some answers. An appropriately planned set of experiments is likely to produce more information than random tests, so we have devoted some effort toward generating a plan. This plan is briefly outlined in the next section.

## B. The Plan

We have divided the experimental program into three major parts called Phases A, B, and C. In Phase A, missions all require the automaton to go from its present position to some explicitly named goal position. The experiments in this phase vary only in the sophistication of the processes used by the automaton to complete each mission.

In Phase B, the mission is still to go to a goal position, but now the goal is only implicitly specified. In general the automaton will have to do some problem-solving and/or actual physical exploration before the goal position is known explicitly. Once the goal position is known explicitly, the mission can be completed by using the techniques developed in Phase A.

In Phase C, the automaton missions will involve simple manipulations of the world--for example, changing the location of objects. In each class of experiment we will be attempting to identify those automaton system strategies that lead to optimal performance of the given missions.

### 1. Phase A

While it is impossible to describe precisely all of the experiments that will be done, since each will depend in part on the results of the previous ones, we plan three categories of experiments in Phase A where the mission is to go to an explicitly named goal. These three categories can be defined roughly as follows:

Category 1--In these experiments, the automaton will be blind and will make use only of its touch sensors. It will store no model of the world and will take actions that can be described as simple reflex responses that are functions of its present state, its touch inputs, and the goal position. These experiments employ programs similar to GOTO1 described in the Third Interim Report (pp. 29-30).

We have already performed some of these experiments for the purposes of testing various low-level components of the system.

Category 2--These experiments, using programs similar to GOTO2 described in the Third Interim Report (p. 30), use the touch and range-finder sensors (but not vision) to build a model. This model is then used to calculate a route to the goal. Here we can begin exploring questions about information-gathering strategies and the relative weights to be attached to unknown areas.

Category 3--Here is the first time that we will be using vision. The visual system as well as touch and range-finder information will all contribute to the model. Just as in the Category 2 experiments, the model will be used to calculate a route. These experiments will use programs similar to GOTO3 described on pp. 30-31 of the Third Interim Report.

## 2. Phase B

In the experiments of Phase B, the primary mission is still to go to a specified place, but now this place is only specified implicitly. The following commands are examples containing implicitly specified places:

- (1) Go to a doorway.
- (2) Go to a place half-way between the two small cubes.
- (3) Go to a place from which all objects in the room are visible.

To execute commands such as these, the automaton system will first have to do some problem-solving in order to convert the implicitly described place into an explicit coordinate pair. In some cases the problem can be solved by referring only to information already in the model; otherwise, a subgoal to obtain additional information will be

set up. This subgoal itself implicitly specifies places that must be visited, so that it is possible to conceive of quite difficult experiments during this phase. Some approaches toward problems of this type were discussed in the Third Interim Report (pp. 31-35).

We will need a visual pattern-recognition capability for the experiments in this phase. Simple programs have been written to allow discrimination between cubes and objects of other shapes, so that we could, for example, give the automaton the mission: Go to a cube.

The ability to recognize shapes adds new dimensions to model building. During Phase A, model building consisted only of noting whether certain grid cells were full or empty. Now that we can classify the objects that fill grid cells, a host of questions emerge that can only be attacked after considerable experience with various alternatives. For example, should the automaton system invent its own categories for objects, or instead attempt to fit everything into those categories that the experimenter might have furnished?

### 3. Phase C

In these experiments we will be combining the abilities developed in Phases A and B to achieve more complex missions involving the manipulation of objects in the environment. Although we are not planning to add an arm to the vehicle, the automaton will be able to move objects about on the floor by pushing them. Some example missions in increasing order of difficulty might be:

- (1) Move Object No. G03071<sup>\*</sup> from its present position at  $(x_1, y_1)$  to  $(x_2, y_2)$ .
- (2) Move a cube to position  $(x_2, y_2)$ .
- (3) Move a cube next to a wedge.
- (4) Move all of the cubes into the northwest corner of the room.

---

<sup>\*</sup>The internal name of an object.

- (5) Build a "wall" of cubes across the room with a gap in it large enough to pass through.

Although in all probability we shall not be experimenting with any but the simplest of these missions in the present project, we shall nevertheless be able to define much more clearly the techniques needed to achieve some of the more difficult missions. Obviously, we can construct missions of this sort having arbitrarily high complexity. How many of these missions we shall be performing depends to a large extent on the problems we encounter during the experiments of Phases A and B.

**Appendix**

**DATA-COMMUNICATION-INTERFACE LOGIC**

## Appendix

### DATA-COMMUNICATION-INTERFACE LOGIC

by

J. A. Baer

#### 1. General Description

At the present time the mobile vehicle is connected to the fixed computer interface equipment by means of a multi-conductor cable. We are now in the process of replacing this cable with a radio link. The cable carries video information from the TV camera on the vehicle and coded digital data that originates both at the vehicle and at the fixed equipment. It is the latter--the coded digital data--that is of concern here. Additional interface logic is also being connected between the radio link and the remainder of the system in such a way that the overall system functions in the same manner as it does under hard-wire control. The vehicle and fixed equipment operation were initially organized to facilitate operation via a radio link. In fact, the connection of the interface logic into the overall system will be done through the same physical connector receptacles that are currently mated to the cable plug. This will minimize the checkout time required for the installation of this expansion of the system.

In Fig. A-1 the block diagram shows symbolically the relationship between the various portions of the equipment. The Fixed Control Unit (FCU) is a part of the fixed equipment, and it communicates with the computer and, at present, the multiconductor cable. A total of 13 signals are passed from the FCU to the cable; 8 of these signals are used to denote distances to be traveled, addresses, etc. The remaining 5 signals are special control signals, such as Reset. Each of the 13 signals appears on its own terminal pair in the FCU--i.e., the data is presented in parallel. The radio link transmits only one bit of information at any one time so the Data Communication Interface (Fixed), or DCIF, must convert the parallel data into ordered sequential data. This, then, is the primary function of the transmitter portion of the DCIF.

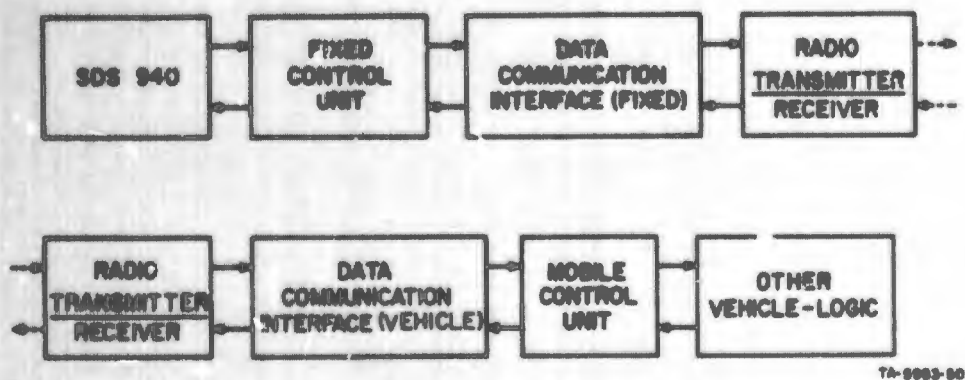


FIG. A-1 INFORMATION-EXCHANGE PATHS

The data exchange between the SDS-940 computer and the FCU is in the form of eight-bit characters; this exchange takes place in an asynchronous manner as controlled by auxiliary signals. These eight-bits require eight time-slots in the sequential data transfer of the radio link. In addition to these eight, the radio link operation, per se, requires three time-slots, so the exchange of information from the radio transmitter to the radio receiver requires eleven time-slots (plus an "off" period between groups of eleven time-slots). Each time slot is about 5 milliseconds in duration (200 bits per second) and is accurately controlled by means of a crystal oscillator.

A functional block diagram of the DCIF is given in Fig. A-2. The top part of the diagram represents the receiver portion of the logic, and the transmitter portion is shown in the bottom part of the diagram. The audio signal input from the radio receiver to the threshold detector is ideally a sine wave of 2083 Hz or 3125 Hz, representing logic ONE and ZERO respectively. The presence of one of these two tones will be detected by the digital filter and, if certain other criteria are satisfied, will pass on the logic ONE or ZERO to the buffer register. The location of the transferred bit in the register is determined by the time-slot generator. The logic and timing sequences in these circuits are arranged such that a sequence of tones at the input causes ONES and ZEROS to be presented in an ordered fashion to the several output terminals.

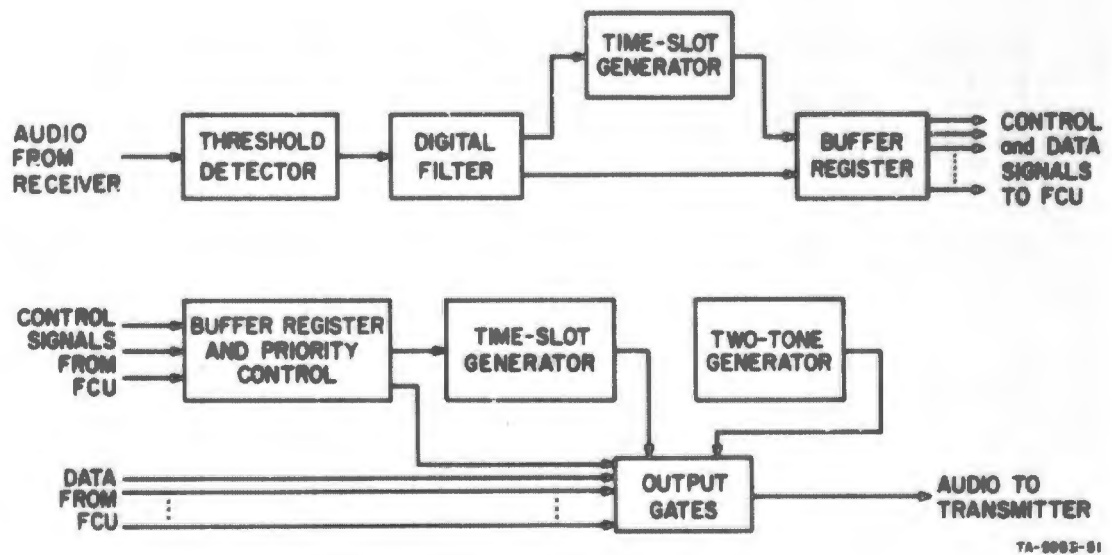


FIG. A-2 DATA-COMMUNICATION INTERFACE (Fixed)

The transmitter portion of the DCIF receives data and control signals from the FCU asynchronously. The control signals, after having the proper priority assigned to them, initiate the time-slot generator. The time-slot generator together with the tone generator and output gates converts the control and data signals into a time sequence comprising two audio tones.

For the interface logic that is on board the vehicle--the Data Communication Interface (Vehicle), or DCIV--the functional block diagram is the same as for the DCIF shown in Fig. A-2, except that the Mobile Control Unit (MCU) receives and sends the data rather than the FCU. There are several differences in the detail of the logic, however, between these two interface logic units. In particular, the buffer register and priority control block in the DCIV is more elaborate than it is in the fixed equipment because there are many interrupt and error signals that must be processed. Because of these interrupts and errors the total number of signals passed from the MCU to the FCU through the radio link is 17 (rather than the 13 that are passed in the other direction).

## 2. Tone Identification

It is not the purpose of this appendix to give a detailed description of the logic in the interface units, but we will discuss here some of the tone-identification processes.

There are two separate crystal-controlled oscillators in the interface system; one is in the DCIF and the other in the DCIV. These operate at 100 kHz and 200 kHz respectively. (The fact that these two oscillators operate at different frequencies is not significant to the system operation; these are the standard units available from the two suppliers of the logic used in the system.) These frequencies are used to establish a time base in the digital filters and are "counted down" to supply clock frequencies for the logic and to supply the audio tone frequencies. These oscillators are not synchronized, so there is a phasing ambiguity that is accounted for in the system timing.

Upon receipt of an audio signal of sufficient amplitude from the radio receiver, a counter is started as the means for measuring the half-period of the incoming audio frequency. When a specified number of half-cycles having the proper time duration have been received, a logic indication is given, signifying that a particular tone has been received. To initiate the receiving cycle in the interface unit, the first tone received must have a frequency of 3125 Hz, and the second tone received must have a frequency of 2083 Hz. The requirements (1) that the frequencies must be present for a certain length of time and (2) that the first two tones appear in a certain order, along with the squelch control in the radio receiver, provide "start-up" protection against random signals.

After the receive cycle in the logic has been initiated, there are two additional protective criteria that must be satisfied. The first criterion relates to the timing of the received audio tones. Upon initiation of the receive cycle, the time-slot generator is started. During each of the eight succeeding time-slots, the logic requires that one or the other of the two tones be detected by the digital filter. If this tone sequence is not received, an error indication is generated,

and the logic reverts to a quiescent state awaiting receipt of a "start-up" condition. If the cause of the failure to receive the tone sequence is loss of RF energy from the fixed-equipment transmitter, the vehicle power is automatically turned off.

The second criterion that must be satisfied is a parity check on the data that is passed through the interface units. If the parity condition is violated, an error indication is generated.

There are additional criteria that could be implemented to provide additional reliability to the radio link. This, naturally, means greater cost and complexity. We are implementing a system that, in our judgment, has a reasonable balance between cost/complexity and reliability, as they relate to the entire automaton system.

#### REFERENCES

1. C. A. Rosen and N. J. Nilsson (Eds.), "Application of Intelligent Automata to Reconnaissance," First Interim Report, Contract AF30(602)-4147, RADC-TR-66-822 (November 1966). AD 817 189.
2. C. A. Rosen and N. J. Nilsson (Eds.), "Application of Intelligent Automata to Reconnaissance," Second Interim Report, Contract AF30(602)-4147, RADC-TR-67-171 (March 1967). AD 820 989.
3. C. A. Rosen and N. J. Nilsson (Eds.), "Application of Intelligent Automata to Reconnaissance," Third Interim Report, Contract AF30(602)-4147. RADC-TR-67-657 (December 1967). AD 827 938.
4. C. A. Rosen and N. J. Nilsson, "An Intelligent Automaton," IEEE International Convention Record, Part 9 (1967).
5. B. Raphael, "Programming a Robot," paper to be presented at the IFIPS Congress 68, Edinburgh, Scotland, 5-10 August 1968.
6. G. E. Forsen, "Processing Visual Data with an Automaton Eye," in Pictorial Pattern Recognition (Thompson Book Co., Washington, D.C., 1968).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Stanford Research Institute Menlo Park, California		2a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	
		2b. GROUP N/A	
3. REPORT TITLE APPLICATION OF INTELLIGENT AUTOMATA TO RECONNAISSANCE			
4. DESCRIPTIVE NOTES (Type of report and inclusive dates) Fourth Interim Report Covering the period 18 December 1967 to 5 April 1968			
5. AUTHOR(S) (First name, middle initial, last name) Nils J. Nilsson Bertram Raphael Sven E. Wahlstrom			
6. REPORT DATE June 1968		7a. TOTAL NO. OF PAGES 33	7b. NO. OF REFS 6
8a. CONTRACT OR GRANT NO. AF 30(602)-4147		9a. ORIGINATOR'S REPORT NUMBER(S) Fourth Interim Report SRI Project 5953	
8b. PROJECT NO. 4594		9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) RADC-TR-68-196	
8c. Task 459405			
10. DISTRIBUTION STATEMENT This document is subject to special export controls and each transmittal to foreign governments or foreign nationals or their representatives may be made only with prior approval of RADC (EMIRC), Griffiss Air Force Base, New York 13440.			
11. SUPPLEMENTARY NOTES RADC Project Engineer: Miss Patricia Langendorf		12. SPONSORING MILITARY ACTIVITY Rome Air Development Center (EMIRC) Griffiss Air Force Base, New York 13440	
13. ABSTRACT This report describes the results of research during the past four months on the project, "Applications of Intelligent Automata to Reconnaissance." The primary goal of this project is to investigate techniques in artificial intelligence applied to the control of a mobile automaton in a real environment. The main emphasis is on the design of a hierarchy of computer programs that will accept visual and other sensory information gathered by the automaton and will direct the actions of the automaton in performing missions that require the abilities to plan ahead and to learn from previous experience.  The work described has largely been aimed toward preparing for an extensive series of experiments with the automaton system, and completes all of the items specified under Phase II of the project. This report summarizes our present status in software and hardware development and then outlines the purposes and plan for the experiments to follow.			

DD FORM 1 NOV 68 1473 (PAGE 1)

S/N 0101-807-6801

UNCLASSIFIED  
Security Classification

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Artificial Intelligence						
Pattern Recognition						
Automata						
Modeling						
Visual Perception						