

**FAA-74-20**  
REPORT NO. FAA-RD-75-65

**THE DEVELOPMENT OF A SIGNAL DATA CONVERTER  
FOR AN AIRPORT VISIBILITY MEASURING SYSTEM**

Hector C. Ingrao  
Melvin Yaffee  
Michael F. Cartwright  
Paul Madden  
Mukund Desai  
Glenn Mamom



AUGUST 1975  
FINAL REPORT

DOCUMENT IS AVAILABLE TO THE PUBLIC  
THROUGH THE NATIONAL TECHNICAL  
INFORMATION SERVICE, SPRINGFIELD,  
VIRGINIA 22161

Prepared for  
U.S. DEPARTMENT OF TRANSPORTATION  
FEDERAL AVIATION ADMINISTRATION  
Systems Research and Development Service  
Washington Dc 20591

**NOTICE**

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

**NOTICE**

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

1. Report No. FAA-RD-75-65		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle THE DEVELOPMENT OF A SIGNAL DATA CONVERTER FOR AN AIRPORT VISIBILITY MEASURING SYSTEM				5. Report Date August 1975	
				6. Performing Organization Code	
7. Author(s) Hector C. Ingrao,* Melvin Yaffee,* Michael F. Cartwright,* Paul Madden,** Mukund Desai,** Glenn Mamom**				8. Performing Organization Report No. DOT-TSC-FAA-74-20	
9. Performing Organization Name and Address U.S. Department of Transportation Transportation Systems Center Kendall Square Cambridge MA 02142				10. Work Unit No. (TRAIS) FA415/R6121	
				11. Contract or Grant No.	
12. Sponsoring Agency Name and Address U.S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington DC 20591				13. Type of Report and Period Covered Final Report July 1973 to June 1974	
				14. Sponsoring Agency Code	
15. Supplementary Notes * Transportation Systems Center, Optical Devices Section, Cambridge MA. ** The Charles Stark Draper Laboratory Inc., Cambridge MA.					
16. Abstract <p>The Optical Devices Group at the Transportation Systems Center has been involved in the development of a breadboard Airport Visibility System (ARVIS) for the FAA since FY 72. One major subsystem in the ARVIS is the Signal Data Converter whose characteristics were initially identified in a report (DOT-TSC-FAA-72-1) titled "Characters of a Signal Data Converter for a Multi-Runway Visibility Measuring System," October 1971. Various aspects relative to the determination of RVR have been reviewed and efficient algorithms developed for the computation of RVR from Allard's and Koschmeider's Law. A sixteen bit wordlength has been established as necessary to provide adequate range and accuracy in the determination of RVR.</p> <p>A breadboard ARVIS was designed and built. Software was developed and parameters representative of various airport operational situations synthesized, exercised and verified, adequately demonstrating the feasibility and versatility of the proposed ARVIS. There remains the ARVIS field testing.</p>					
17. Key Words Visibility Transmissometer Runway Visual Range Air Traffic Control			18. Distribution Statement  DOCUMENT IS AVAILABLE TO THE PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161		
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 226	22. Price





## PREFACE

This report briefly describes the evolution of the FAA/NBS RVR transmissometer system into a breadboard Airport Visibility Measuring System (ARVIS) which has been laboratory tested and which will undergo field tests at the National Aviation Facilities Experimental Center (NAFEC) in Atlantic City, NJ during 1975.

Appendix I of this report documents in detail the development of a Signal Data Converter Unit (SDCU) as a replacement for the SDCU used in the FAA/NBS RVR system.

Appendix II describes an analog computer for calculating RVR. Appendix II was prepared by Joseph Horner, TSC.



## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. INTRODUCTION.....	1-1
2. AN EVOLUTION FROM THE PRESENT FAA RVR SYSTEM TO AN AIRPORT VISIBILITY MEASURING SYSTEM.....	2-1
3. PRESENT FAA SYSTEM FOR RVR MEASUREMENTS.....	3-1
3.1 Introduction.....	3-1
3.2 Transmissometer.....	3-1
3.3 RVR Computer.....	3-3
3.4 RVR Remote Digital Display.....	3-4
3.5 RVV Strip Chart Recorder and Indicator.....	3-5
4. PROPOSED VISIBILITY MEASURING SYSTEM.....	4-1
4.1 System Approach.....	4-1
4.2 Visibility Information.....	4-6
4.3 Visibility and Background Information as Control Parameters for Intensity Setting of Runway Lights.....	4-7
5. PROGRESS ON THE VISIBILITY MEASURING SYSTEM.....	5-1
5.1 250-Foot Baseline Transmissometer, Modification I.....	5-1
5.2 75-Foot Baseline Transmissometer, Modification III.....	5-3
5.3 Central Processor and Control Unit.....	5-5
5.4 Signal Data Converter Unit.....	5-8
REFERENCES.....	R-1
APPENDIX I - STUDY DESIGN AND IMPLEMENTATION OF A SIGNAL DATA CONVERTER AND SIGNAL SIMULATOR FOR A VISIBILITY MEASURING SYSTEM.....	I-i
APPENDIX II - ANALOG COMPUTER FOR CALCULATING RVR.....	II-i

## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
2-1.	Block Diagram of an Evolution from the Present FAA RVR System to an Airport Visibility System (ARVIS).....	2-2
4-1.	Block Diagram of the Experimental ARVIS Developed at TSC (MOD III).....	4-5
5-1.	Layout of the Central Processor and Control Unit (CPCU) of the TSC Airport Visibility System (ARVIS)....	5-2
5-2.	Photograph of the Central Processor and Control Unit (CPCU) of the TSC Airport Visibility System (ARVIS)....	5-6

LIST OF TABLES

<u>Table</u>	<u>Page</u>
4-1. REMOVAL OF FAA/NBS RVR SYSTEM UNITS AND CORRESPONDING REPLACEMENTS AND/OR ADDITIONS IN PROPOSED TSC MODIFICATIONS.....	4-2

## LIST OF SYMBOLS AND ABBREVIATIONS

<u>Symbol</u>	<u>Definition</u>
ALCH	approach light contact light
ARVIS	airport visibility system
CPCU	central processor and control unit
CSDL	Charles Stark Draper Laboratory, Inc.
ILS	instrument landing system
LED	Light emitting diode
MOD I	modification I
MOD II	modification II
MOD III	modification III
MOD IV	modification IV
NAFEC	National Aviation Facilities Experimental Center
RVR	runway visual range
RVV	runway visibility value
SDCU	signal data converter unit
SVR	slant visual range
TVR	taxi visual range

## 1. INTRODUCTION

The Transportation Systems Center (TSC) of the Department of Transportation is engaged in a continuing program for the Federal Aviation Administration titled "Airport Visibility Measuring Systems".

This report, in compliance with the Project Plan Agreement (PPA) FAA-515, summarizes the results and recommendations on one of the tasks under the program. The specific task reads as follows:

"Based on the expectation that newer systems will evolve for the measurement of RVR and probably SVR and that instrumentation at multi-runway airports will consist of numerous and varied installations and that the periodicity of update of the observables will increase, the inadequacy of present day signal data converter equipment is apparent. To this end a set of performance characteristics for an airport visibility SIGNAL DATA CONVERTER UNIT (SDCU) shall be developed. The display of the observables shall govern the establishment of SDCU characteristics to the extent that sufficient visibility information shall be available for presentation of operating minima associated with landing categories through CAT III."

TSC has developed and laboratory tested a SDCU (Reference 1) under contract\* to the C.S. Draper Laboratory (see Appendix I). It has prepared also a specification to facilitate the procurement of an engineering prototype model for an operational test program and submitted it under different cover. It suggested that the procurement of an SDCU be made as part of an ARVIS including a modified 75-foot baseline transmissometer.

---

\*DOT/TSC 460





## 2. AN EVOLUTION FROM THE PRESENT FAA RVR SYSTEM TO AN AIRPORT VISIBILITY MEASURING SYSTEM

An earlier report (Reference 2) summarizes the TSC study which resulted in the definition of characteristics for an SDCU for use in airport visibility measuring systems. The study objective was stated as follows:

"Characteristics will be defined for a signal data converter for computing visibility values from inputs from several transmissometers with reference to several kinds of target lights (e.g., centerline lights, approach lights, edge lights, taxiing lights)."

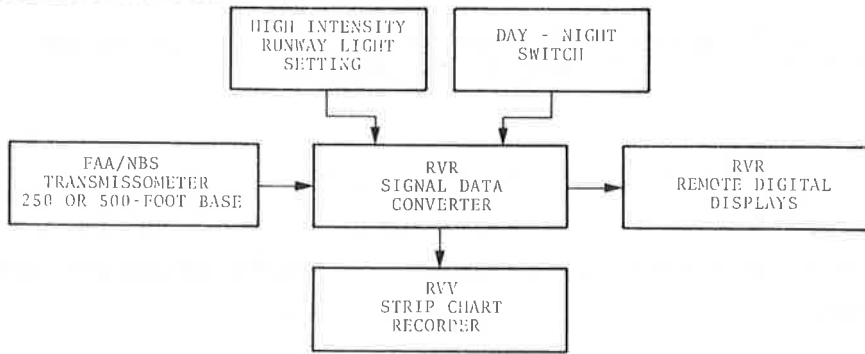
Economical and operational considerations led the TSC work to an evolutionary process from the present FAA RVR system to an ARVIS. The full system is expected to be reached after a 4-step modification process (Figure 2-1).

The first step in this modification (MOD I) consists of the modernization of present transmissometers (projector power supply and receiver) by using solid state circuitry and components. The second step (MOD II) will consist of, in addition to MOD I, the substitution of the present RVR computer with a SDCU and teletype with the capability of handling the simultaneous signals from several transmissometers distributed along runways.

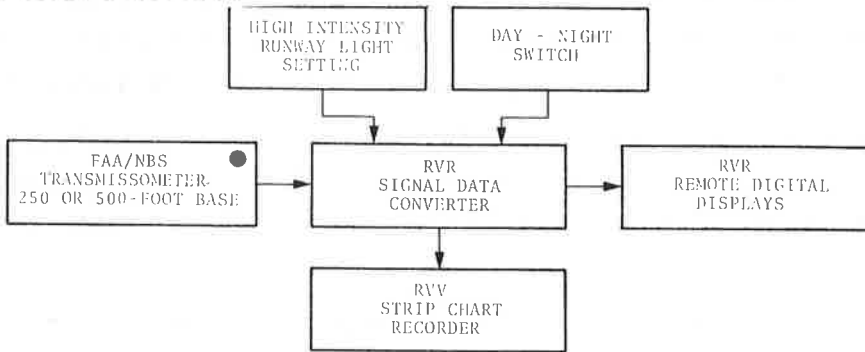
The third modification (MOD III) implies a system approach to the airport visibility measurements and reporting. By considering the airport as the system, all visibility measuring sensors in the airport, all light systems used as visual cues, and sky background luminance sensor are integrated in a true ARVIS. The ARVIS is a software oriented system in which performance characteristics (frequency of RVR updating, different processing of visibility data, modification of display data in accordance to specific airport needs, etc.) can be changed without changes in hardware.

The implementation of MOD III consists of the expansion of the MOD II SDCU to a Central Processor and Control Unit (CPCU), the replacement of the MOD I (or MOD II) receiver for one with the

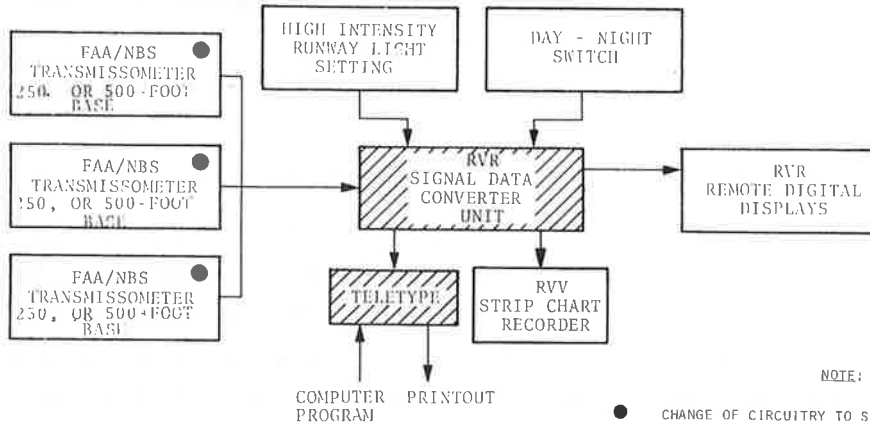
PRESENT FAA RVR MEASURING SYSTEM



1ST MODIFICATION (MOD I)



2ND MODIFICATION (MOD II)



NOTE:

- CHANGE OF CIRCUITRY TO SOLID STATE COMPONENTS
- EITHER 75, 250 OR 500 FOOT BASIS TRANSMISSOMETER OR IN ANY COMBINATION
- ▲ THE SAME AS WITH SDCU IN THE 2ND MODIFICATION BUT WITH ADDITIONAL MODULES FOR ADDED INPUTS AND/OR FUNCTIONS
- ▨ ADDITIONS, SUBTRACTIONS, SUBSTITUTIONS AND/OR MODIFICATIONS TO PRESENT FAA RVR MEASURING SYSTEM COMPONENTS

Figure 2-1. Block Diagram of an Evolution from the Present FAA RVR System to an Airport Visibility System (ARVIS)

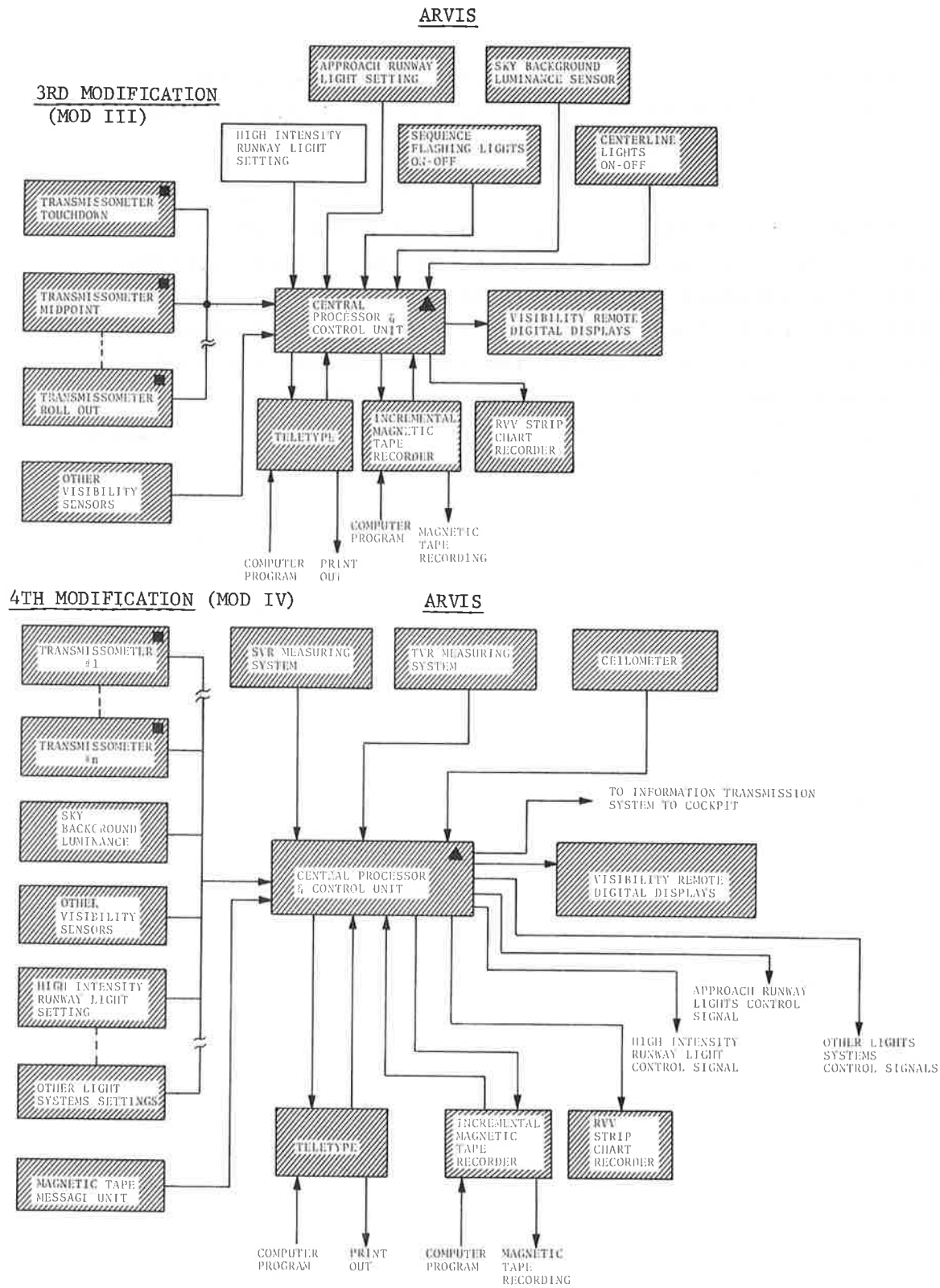


Figure 2-1. (Continued)

capability of internal calibration and larger dynamic range for CAT I, II and III operation, and a slave control which is part of the ARVIS control and failure monitoring system.

The fourth modification (MOD IV) consists of a more comprehensive expansion of the ARVIS MOD III system with the inclusion of SVR and TVR data, automatic control of the airport lighting settings in accordance with the visibility conditions, and automatic transmission to the pilot of the visibility information required. This automatic transmission will eliminate the burden on the controller to relay this terminal information to the pilot. Nevertheless, the controller will be in parallel with this information channel and will have the possibility to override it, if required.

### 3. PRESENT FAA SYSTEM FOR RVR MEASUREMENTS

#### 3.1 INTRODUCTION

The present procedure used by the FAA to determine visibility at airports depends heavily on the transmissometer as an instrument to complement and, at times, replace human observers. The evolution to a comprehensive airport visibility measuring system (ARVIS), as discussed herein, will likewise center on the transmissometer as the basic sensor for gathering atmospheric transmittance.

In this section a brief description is given of the entire FAA/NBS RVR system now in use. The system includes the transmissometer, the RVR computer, the RVR remote digital displays and the RVV strip recorder and indicator.

#### 3.2 TRANSMISSOMETER

The transmissometer, first developed by Douglas and Young at the National Bureau of Standards in 1942, measures the atmospheric transmittance over a fixed distance (usually 250 feet) with a light source and a photo-detector at opposite ends of the sampled path. First accepted for airport operations in 1952, the transmissometer serves to measure RVR or RVV along more than 270 runways. Although it has been modified by the addition of heaters, blowers, power stabilizers, etc., the basic design and operating principles of the instrument have not been changed since the first transmissometers were installed. The "Preliminary Instruction Book - Runway Visual Range (RVR) System" (Reference 3) prepared for the FAA gives a concise description of the transmissometer. The manual describes it as follows:

"The transmissometer measures atmospheric transmission by projecting a well collimated beam of light down a base line installed near the ILS glide slope transmitter building or adjacent to the touchdown area of the ILS runway, and detecting the intensity of this light in a photoelectric receiver located at the opposite end of the baseline. The receiver translates the

intensity of the received light into a pulse rate by using photo-electric current generated in a vacuum photo-electric cell to charge a capacitor. When a given charge accumulates on the capacitor, a gas discharge trigger tube connected across the capacitor breaks down delivering a large impulse to the following circuitry and reducing the voltage of the capacitor to a low value equal to the extinction voltage of the gas discharge tube. This process is repeated; the time required to accumulate this charge being inversely proportional to the photocurrent and hence to the light intensity. Thus, the pulsing frequency of the circuit is linearly related to the light intensity. Through the use of an iris diaphragm in the optical system of the receiver, this pulsing rate is adjusted to 4,000 pulses per minute for 100% transmission, i.e., a clear day free of smoke, dust or haze, in the baseline path. Any such aerosol, including fog or rain, in the baseline path reduces this light intensity and hence the pulsing rate by absorbing or scattering light from the beam. Ideally, no extraneous sources of light should be permitted to enter the optical system of the receiver such that the pulsing rate would be zero in the absence of a beam from the transmitter. In any actual situation a background level of illumination exists necessitating subtracting the pulse rate measured with the transmitter source off and the pulse rate measured with the transmitter source on. This background correction must be performed from time to time to take into account changing sun position, sky brightness or weather conditions which result in spurious light scattered into the receiver. This background correction is performed manually by either switching off the projector from the indicator front panel switch and observing the recording milliammeter indication, or initiating a background check sequence by pressing a button on the signal data converter power supply and control chassis or any remote indicator chassis connected to the signal data converter control and power supply. More than one transmissometer may be used per runway as required for operation at runways approved for lower visibility operation."

### 3.3 RVR COMPUTER

The same manual continues: "The signal data converter computer contains the necessary time base, clock dividers and counters to permit obtaining a digital value for the transmissometer output. A separate counter is used to count and store the background count which is subtracted from the normal transmission count by entering the complement of the background count into the transmission counter prior to the 45 second period over which the transmissometer output is counted. Transmissometer output is counted for a 45 second period and then transferred into a static storage register. Three seconds later, the transmission counter is cleared, the background complement entered, and the process repeated. The value of transmissivity obtained through this count is stored such that a computation of the RVR value can take place at any time. While under normal conditions, a computation of RVR takes place only once in 48 seconds, a recomputation is initiated whenever a different RVR table is selected in response to a change in runway light setting or a change in the status of the day/night switch. These two inputs serve to select one of the six RVR tables which are plugged into the signal data converter. Systems are furnished either with class I tables pertaining to a 500 foot base line or class II tables pertaining to a 250 foot base line for the transmissometer. With a table selected, the RVR value is obtained from the table by applying clock pulses to the input of the transmissivity storage register causing it to count upward from the value of transmissivity previously stored. The number of pulses supplied to the transmissivity register is precisely equal to the capacity of the register, 2048 pulses. Thus, at the end of such a counting-compute cycle, the value stored in the transmissivity register is exactly the count which was originally stored in this register. The output of the transmissivity register is translated into a hexadecimal code and applied to the selected RVR table, an

output pulse is obtained from the selected RVR table, meaning that the RVR value exceeds or is equal to the value represented by such a count. This pulse output from the RVR tables is passed through a gate and into a five bit counter. The purpose of the gate is to prevent any pulses from reaching the five bit counter until the transmissivity register has passed the overflow point; thus the number of pulses entering the five bit counter is equal to the number of RVR values which are passed between the time the transmissivity register overflows, and hence reads zero, and the time it counts up to its original stored value. Thus, the count in the five bit counter is equal to the number of the solution from zero to 21 which corresponds to the RVR value to be displayed.

The value of the RVR is transmitted to the Remote Digital Display as discussed below. During the time the SDC is not transmitting an RVR value, the receiver decoder transmits the value of runway light setting received from the runway light intensity relay box in the form of one of three frequencies with which the line is switched to common. This switching of the line is detected by the SDC and used to select the particular table required depending on the status of the day/night switch. Up to five remote indicators or computer selectors can be connected to either a signal data converter or a receiver decoder. Additional features of the signal data converter computer include two test provisions, one of which substitutes a crystal clock frequency for the transmissometer pulse output, and the other cycles the indicators through all the possible RVR values. In order to test all tables, a manual table select is available in conjunction with the first test."

#### 3.4 RVR REMOTE DIGITAL DISPLAY

The RVR data is distributed from the RVR computer to the controllers in the control tower as numerical readouts on the Remote Digital Display Unit and then via voice link to the pilot in the aircraft. The RVR display at Logan International Airport, Boston, e.g., has the capacity to monitor two runways at a time (one transmissometer each), as selected by runway selector switches.



Readings range from 1000 to 6000 feet\* and are updated approximately once a minute. RVR is given in hundreds of feet by the first two digits on the display with the third digit indicating a "+" (in excess of 6000 ft) or a "-" (less than 1000 ft).

Under conditions not requiring RVR computation, i.e., when the airport's runway lights are in intensity positions "1" or "2", an "L" appears in the third digit position of the display, indicating that the RVR computer is inoperative because it is not required.

The signals to the display unit are derived from the SDC as follows. The contents of the five bit counter holding the value of the currently-calculated RVR are decoded to yield a signal on one of the 21 lines corresponding to the 21 solutions. These lines are re-encoded into a modified indicator code which is used to operate a bank of nine relays, three for each digit and three for the symbol following the two digits. The relay contacts are appropriately wired to route the proper positive and negative voltages to the proper indicator terminals in order to display the appropriate numbers. The indicators remain quiescent until a solution is obtained at which time they are strobed for 0.75 seconds to display the new value. Simultaneous with the strobing of the indicators, the nine bit modified indicator code is transmitted in serial binary to the receiver decoder along with a parity bit for error detection. If a parity error is detected or the line is interrupted, the receiver decoder automatically forces the display to read "\_\_\_ \_\_\_ E."

### 3.5 RVV STRIP CHART RECORDER AND INDICATOR

The pulse train from the transmissometer is also monitored by the RVV indicator. This indicator is essentially a frequency meter that converts the pulse train signal into a direct current

\*This range is for a 500-foot baseline transmissometer. For 250-foot baseline the lower RVR is 500 feet.

whose magnitude is proportional to the pulse rate. A strip-chart recorder provides a continuous record of the indicator output which is proportional to the instantaneous atmospheric transmittance.

## 4. PROPOSED VISIBILITY MEASURING SYSTEM

### 4.1 SYSTEM APPROACH

The present FAA/NBS RVR instrumentation has well served its function of gathering visibility data, computing RVR values and disseminating the information. It is clear, however, that future demands of ever increasing traffic, lowered landing minima and extensive automation of the landing process and information dissemination will require new approaches to the entire airport visibility measuring techniques. One of the approaches foreseen is the evolution of the present RVR instrumentation into a comprehensive system, increasing its accuracy, adding flexibility using a software approach, and generally improving the quality of the disseminated information as well as its output rate. The criteria to have a software oriented system precluded the consideration of analog computers as the data processing hardware. Nevertheless, for some simple instrument installations, analog computers to calculate RVR should be considered (see Appendix II). The ultimate goal is to monitor and measure the visibility in all the runways and taxiways of an airport using system concepts and state-of-the-art equipment. TSC has proposed the development of an ARVIS to satisfy this goal.

In the interest of continuity, to ensure that changes in the existing visibility instrumentation will not compromise airport safety or efficiency and to introduce changes as required, it is proposed that these be made as a series of successive modifications of the present FAA/NBS RVR system as suggested in Section 2.

The removal of units from the FAA/NBS transmissometers and the replacement and/or addition with corresponding units in the four step modification plan proposed by TSC is summarized in Table 4-1. To avoid confusion and for identification purposes, each unit of the proposed modifications has been identified with a unit number. The data given in Table 4-1 with the information supplied in figure 2-1 is self-explanatory for modifications 1 and 2.

TABLE 4-1. REMOVAL OF FAA/NBS RVR SYSTEM UNITS AND CORRESPONDING REPLACEMENTS AND/OR ADDITIONS IN PROPOSED TSC MODIFICATIONS

Modification	FAA/NBS System Unit to be Removed <sup>†</sup>	TSC Modification Unit to be Installed
I <sup>††</sup>	Pulse Generator Receiver Amplifier - Power Supply (A100-L)	*Receiver No. 10-R or No. 10-R-500
	Projector Power Supply (A300-1)	Projector Power Supply and Control No. 12-P
II	Pulse Generator Receiver Amplifier - Power Supply (A100-L)	*Receiver No. 10-R or No. 10-R-500
	Projector Power Supply (A300-1)	Projector Power Supply and Control No. 12-P
	Signal Data Converter Control and Power Supply	Minicomputer No. 24-C; I/O Interface No. 26-1; Teletype No. 28-T
	Pulse Generator Receiver Amplifier - Power Supply (A100-L)	**Receiver No. 30-R-250 or No. 30-R-75
	Projector Power Supply (A300-1)	**Projector Power Supply and Control No. 12-P
	---	**Slave Control No. 32-S
	Day/Night Switch	Sky Background Luminous Sensor No. 34-L
III (ARVIS) <sup>†††</sup>	Signal Data Converter Control and Power Supply	Minicomputer No. 24-C; I/O Interface No. 35-1; Teletype No. 28-T
	---	Incremental Digital Tape Recorder No. 36-R

TABLE 4-1. REMOVAL OF FAA/NBS RVR SYSTEM UNITS AND CORRESPONDING REPLACEMENTS AND/OR ADDITIONS IN PROPOSED TSC MODIFICATIONS (Continued)

Modification	FAA/NBS System Unit to be Removed†	TSC Modification Unit to be Installed
	---	Photometric Display No. 38-P
	Remote Display	***Remote Digital RVR Display No. 39-D
	RVV Recorder A-400	Strip Chart Recorder No. 31-R
IV (ARVIS)	Expansion of MOD III to satisfy future airport operational requirements. Hardware is not identified as yet.	

†Information on RVR system units is given in Reference 3.

††Hardware description of units and field tests described in TSC report (in preparation).

†††Hardware description in TSC report (in preparation).

\*The receiver No. 10-R-250 or No. 10-R-500 can be modified to operate in a 75-foot base transmissometer by introducing minor optical modifications. (This receiver is designated No. 10-R-75.)

\*\*Number of units depends on number of transmissometers modified.

\*\*\*Number of displays as airport operations requires.

The third modification (see Figure 2-1), MOD III, a true ARVIS, will be the extension of the transmissometer network to three transmissometers per runway (at touchdown, midpoint and rollout, and also other visibility sensors as required, for example, in the approach zone at those airports where shoreline geography creates rapidly changing conditions.) The number of transmissometers and other sensors described in MOD III implies the capability of the ARVIS, but not necessarily the actual deployment of the sensors. The ARVIS will be configured in accordance with the specific airport needs. In the MOD III, it is likely that a shorter baseline transmissometer may be introduced, and the SDCU which is part of MOD II will have to handle the outputs of this multi-transmissometer array and other visibility sensors. The SDCU input/output must be compatible with new visibility sensors. Also, it should be capable of driving new visibility remote digital displays, satisfying system control functions, and data logging.

These requirements lead to the expansion of the SDCU by additional units to the Central Processor and Control Unit (CPCU). Table 4-1 identifies the units to be removed, installed, or replaced in the FAA/NBS RVR system to reach the MOD III (ARVIS) level. Figure 4-1 gives a block diagram of a breadboard ARVIS developed at TSC.

Finally, the last modification, MOD IV, of the ARVIS will take into account all the various light targets used for visual cues, such as high intensity runway lights, taxiway lights, centerline runway lights, approach runway lights, and other lighting systems. It is expected that the CPCU will be able to use this information to calculate and display TVR and SVR. Also, there may be a need in the future to determine and display ceiling information.

The operational definition of TVR is not yet certain. Thus the method for determining TVR is not established. In the case of SVR, there may be a need for rather specialized data analysis. It is possible that the CPCU will have to be expanded in the fourth stage of modification to handle these increased data input-output demands. However, there are several minicomputers available on the market today which have expanded memories and modular architecture. It appears, too, that the very near-future will bring

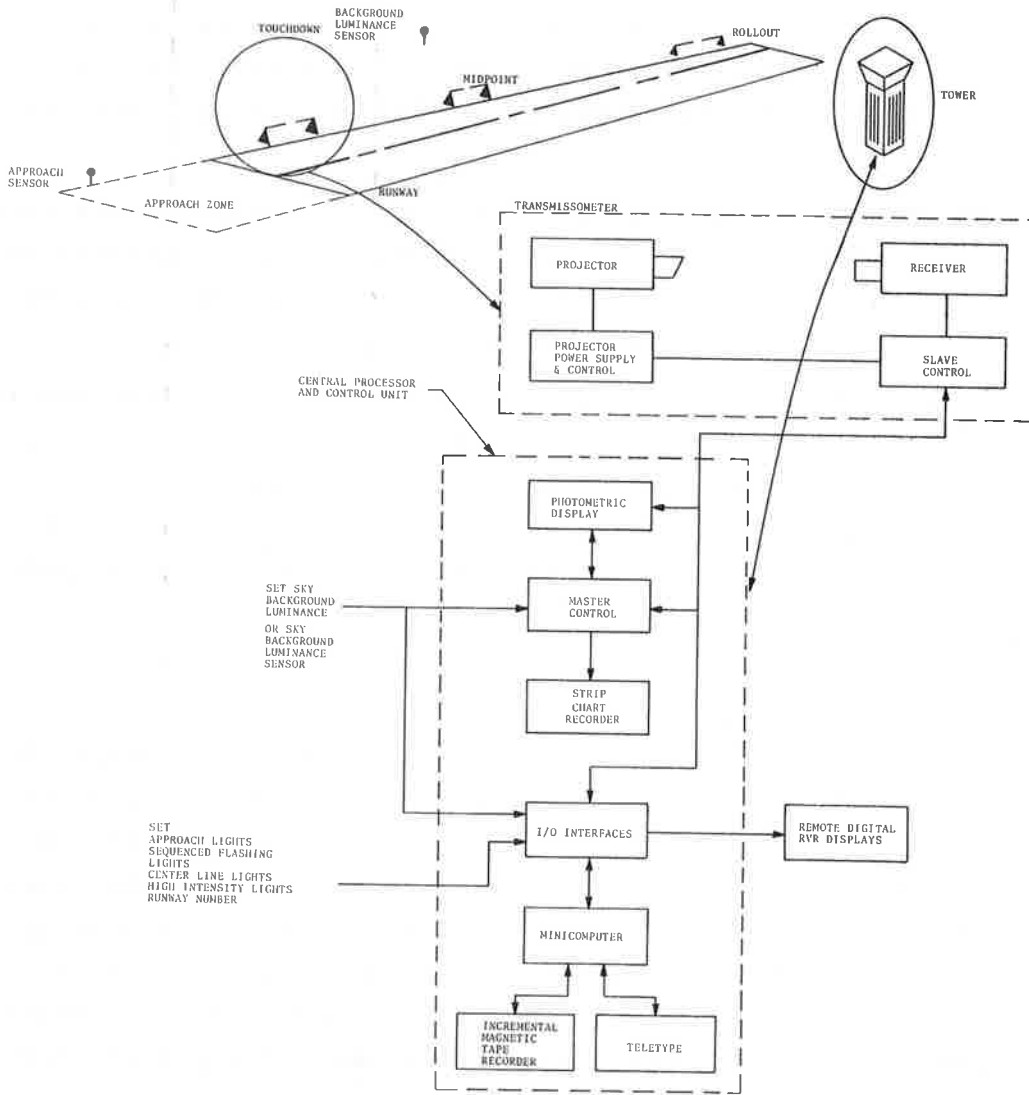


Figure 4-1. Block Diagram of the Experimental ARVIS Developed at TSC (MOD III)

minicomputers with even more capacity, more flexibility and lower prices. Therefore, it is believed that the requirements of this final stage in the evolution of the visibility measuring system can be met by modular architecture and a minimum amount of additional CPCU hardware.

In MOD IV, Figure 2-1, two new functions are shown for the overall visibility measuring system. One is a magnetic tape message system which will send emergency or special message units to the pilot.

The second new feature is the provision for sending visibility information to the cockpit. The scope of this task will depend on the form of cockpit visibility display selected and the data link chosen by the FAA to handle the information.

Due to the complexities of problems to be solved in the MOD IV which have not been defined yet (i.e., TVR, SVR, data link), hardware cannot be precisely identified in this level of ARVIS. Therefore, the purpose of indicating this level of modification is to show that the ARVIS (MOD III) has the capability of being expanded to take care of future operational requirements.

#### 4.2 VISIBILITY INFORMATION

The present operational meaning of "visibility" is simply the RVR; that is, how far along a runway, from the point of touchdown, the runway lights can be seen. This distance, and perhaps also a ceiling indication, represents the only information which the pilot receives about the visual environment into which he is descending. This provides a minimal description of landing visibility which rapidly is becoming inadequate to meet operational needs. A number of improvements have been suggested to complement the present RVR information of describing airport visibility and to enhance the significance of visibility information for the pilot.

As has been described above, an extension of the RVR concept has been proposed in which three transmissometers are used to measure the visibility at three points along the runway. It appears that such installations will give the pilots more information in



patchy fog or non-uniform weather conditions. Almost as important as the knowledge of the variation of visibility along the runway is an indication of how rapidly it is varying. This calls for a method of computation and display which brings attention to significant temporal variations in reported visibility values. Included in any attempt to handle time-varying conditions is the question of desirable up-date intervals. Whereas at present this interval is 53 seconds, the combination of larger planes and lower landing minima (Category II and III) seems to generate a need for faster updating of information. A more effective interval is about 10-12 seconds (see Reference 4).

In addition to RVR, SVR is desirable since it is a measure of how well the pilot can see the approach lights along the glide path. The SVR may well be less than the RVR in cases of inhomogeneous visibility conditions (for example, low ceiling). Since the SVR is the relevant information with regard to pilot orientation during approach, it should be included in any scheme for the modification of the visibility measuring system. The FAA is presently supporting the development of concepts and systems to measure the SVR.

TVR would provide another piece of data for the pilot and controller to use in forming a complete picture of airport visibility. As of this writing, no operational definition exists for this parameter.

Additional visibility information which must be taken into account in characterizing the overall visibility measuring system includes the prevailing visibility (which is dialed in by the controller) and the visibility of sequenced flashing lights and centerline runway lights.

#### 4.3 VISIBILITY AND BACKGROUND INFORMATION AS CONTROL PARAMETERS FOR INTENSITY SETTING OF RUNWAY LIGHTS

At present, the high intensity runway lights are set by steps (usually three) by the controller and in accordance with the visibility and background conditions at the airport. The setting chosen by the controller can be changed on specific request of the

pilot approaching for landing. The pilot communicates via voice with the controller and asks for a reduction or an increase in the intensity of the lights to improve his visibility under the prevailing conditions at that particular time. The same is true for the sequenced flashing lights (on or off).

In the MOD IV proposed visibility data flow system, we suggest the use of the visibility and background information as controlling parameters of a servo system which will control the intensity of the lights for optimum visibility under prevailing conditions. Provisions would be made to allow controller override of the servo system when unusual visibility and/or background conditions as experienced by the pilot warrant such an intervention.

## 5. PROGRESS ON THE VISIBILITY MEASURING SYSTEM

The Optical Devices Section at TSC has been engaged since August 1972 in the development of an experimental system conceptually defined as ARVIS (MOD III) (see Figure 2-1).

One of the design objectives of this breadboard system is that the modifications (III) shall be easily implemented in the field by means of modification units.

Development of the ARVIS system and the corresponding modification units capable of satisfying the 1st, 2nd and 3rd modification proceeded simultaneously.

The layout of the CPCU of the TSC ARVIS is shown in Figure 5-1. TSC is rapidly progressing to the point where field tests of the ARVIS could begin at NAFEC in 1976. The major developments at TSC were those relating to:

1. the modification (MOD I) of a standard FAA/NBS 250-foot baseline transmissometer as indicated in Figure 2-1.
2. the development of a new 75-foot baseline transmissometer as indicated in Figure 2-1 (MOD III configuration).
3. the development of a CPCU for the MOD III configuration, and
4. the development of an SDCU corresponding to the MOD II.

### 5.1 250-FOOT BASELINE TRANSMISSOMETER, MODIFICATION I

In the modification of the standard FAA/NBS 250-foot base (MOD I) transmissometer development by TSC, the original pulse generator, receiver simplifier, power supply (A100-6), and projector power supply (A300-1) were removed from the system. The receiver housing and optics, the projector, the enclosures for the receiver amplifier and the power supply were retained. A new solid state receiver (10-R-250) mechanically interchangeable with the original pulse generator was developed. It was designed to be used in conjunction with the existing RVR signal data. The pulse rate

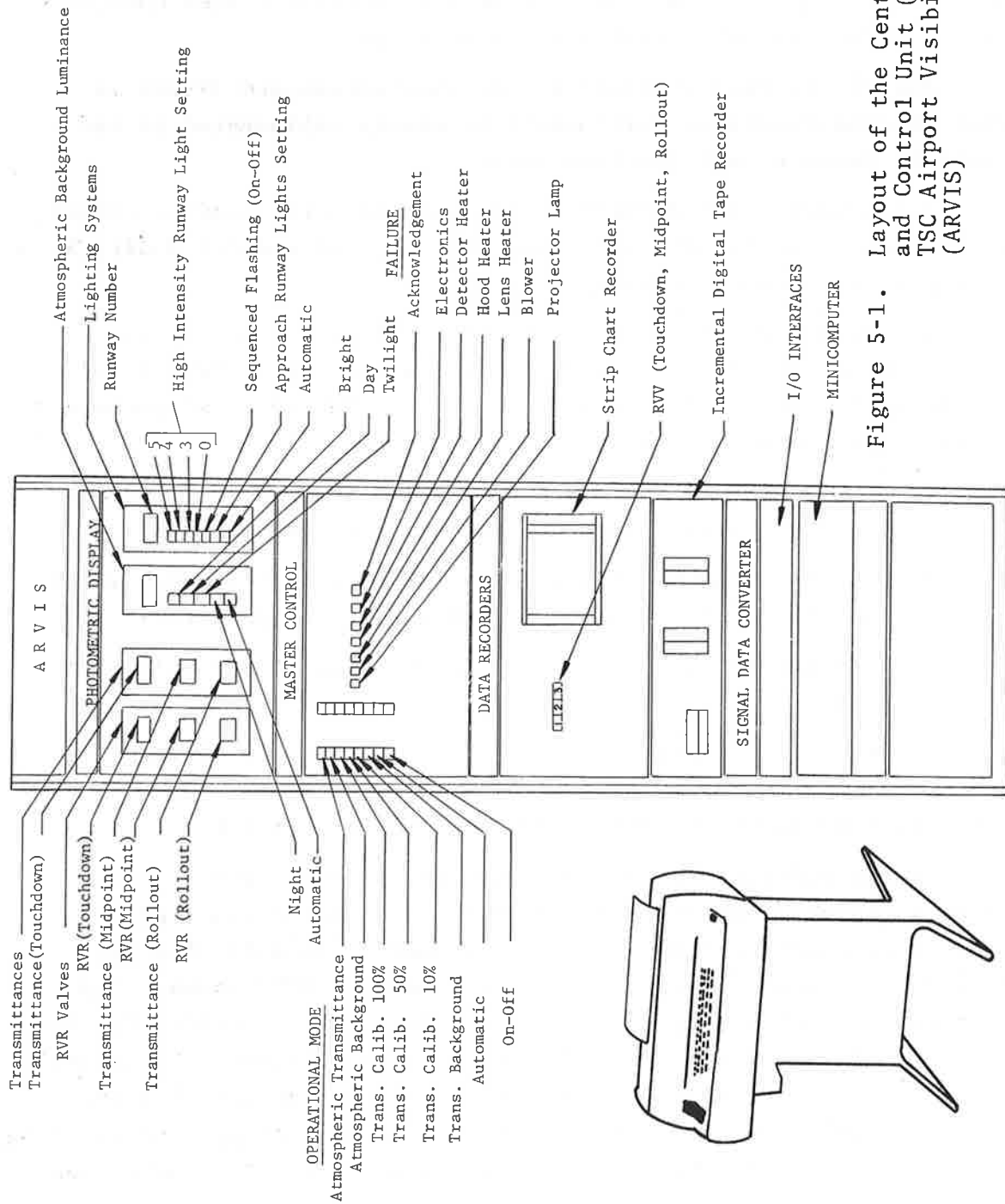


Figure 5-1. Layout of the Central Processor and Control Unit (CPCU) of the TSC Airport Visibility System (ARVIS)

of the 10-R-250 receiver is compatible with the existing RVR computer. An additional design feature of the 10-R-250 receiver is the utilization of a photopic filter ahead of the silicon detector. The filter bandpass was chosen so the detector sees a wavelength spectrum more closely matching the response of the eye of an observer by rejecting a high background level in the near infrared.

It should be indicated that the 10-R-250 can be used in either 500- or 75-foot baseline transmissometers by introducing minor optical modifications. These receivers will be respectively identified as 10-R- 500 and 10-R-75.

The original projector power supply (A300-1) has been replaced with a solid state programmable d.c. power supply and control (12-P) capable of providing stabilized d.c. power to the projector lamp under wide excursions of line input voltage and frequency (105-132 V and 47 to 440 Hz) to facilitate operation under emergency power conditions. In addition, the d.c. voltage for the projector lamp is set at 5 V, increasing considerably the projector lamp life, (lamp nominal rating 6 V), thus reducing system down time and maintenance. To further increase the lifetime of the projector lamp, the 12-P power supply and control maintains a 0.5 V applied to the filament when the transmissometer background is measured. The power supply incorporates additional sensing circuits which facilitate the identification of failure modes in the projector system (power supply and/or lamp filament). The failure mode identification is basic to the MOD III system but is not active in the MOD I or II systems.

## 5.2 75-FOOT BASELINE TRANSMISSOMETER, MODIFICATION III

The 75-foot baseline transmissometer in the ARVIS MOD III system is reached by replacing and/or adding to the FAA/NBS 250-foot base transmissometer the following units: Receiver No. 30-R-75, Projector Power Supply and Control No. 12-B, Slave Control No. 32-S. This 75-foot base transmissometer operates in conjunction with the CPCU described in Section 5.3.

The receiver uses solid state components, has an internal optical calibration system, and failure mode detection circuitry. The No. 30-R-75 receiver measures atmospheric transmittances corresponding to the 6000 - 100 feet RVR range.

The receiver internal calibration functions are exercised periodically and provide optical detection and electronics check by sequencing through several modes: atmospheric transmittance, atmospheric background, calibration of the detector and associated electronics and transmissometer background. This is achievable by modifying the optical path viewed by the detector using a six stage optical turret assembly motor driven by the command of timing circuits in the CPCU. A miniature stabilized incandescent lamp (derated to provide extended life operation in excess of 100,000 hours) is used as the receiver calibration source. Calibration is achieved at 100 percent, 50 percent, and 10 percent equivalent atmospheric transmittance through the use of neutral density filters. The receiver output calibration levels are compared with preset levels in the CPCU to activate failure mode indicators when the calibration levels fall outside a certain tolerance range indicating that corrective maintenance is required.

The other failure modes indicate malfunctions in the receiver heaters, heaters for the optics, the receiver blower, receiver power supply, projector lamp and projector power supply. The projector power supply and control No. 12-B is physically the same used in MOD I and II with a connection difference. The failure mode circuits are connected, via the slave control No. 32-S, to the CPCU. These circuits will indicate failure of the power supply and/or lamp filament.

The slave control No. 32-S receives command signals via a modem from the CPCU to exercise given functions by the receiver and/or power supply and control No. 12-B. Also the No. 32-S transmits via modem to the CPCU data failure signals and operational mode status of the 32-R-75.

### 5.3 CENTRAL PROCESSOR AND CONTROL UNIT (CPCU)

The FAA/NBS RVR System includes the RVR SDCU, RVV Strip Chart Recorder and Indicator in the equipment room at the control tower. The MOD I transmissometer receiver is designed to work with the forgoing RVR SDCU directly. The MOD II System configuration, however, represents a departure from the use of the current RVR SDCU. In their place is a new SDCU integrated by a minicomputer, an I/O interface and a teletype.

In the MOD III the same minicomputer and teletype of MOD II is used, but a new I/O interface (No. 35-I) is introduced. These three components integrate the MOD III SDCU and which is part of the CPCU.

The CPCU (Figure 5-2) is comprised of Master Control, Photometric Display, Data Recorders and Signal Data Converter. The operation of the CPCU is governed by mode selection switches on the Master Control. In the automatic mode an operational sequence is followed and the actual particular mode of operation is verified by the slave control.

In the automatic mode, atmospheric transmittance measurements are made over a 5 minute period followed by a 50 second atmospheric background measurement interval. This sequence is alternately repeated for 10 cycles and is then followed by a maintenance status checking sequence to assure normal transmissometer receiver operation as previously described. The latter sequence is performed during the last minute and 40 seconds of every hour. The time sequence in the CPCU can easily be varied to accommodate airport operational requirements.

The commands are transmitted to the receiver and projector over a two wire telephone line via the Slave Control Unit which is located near the receiver in the fields. The automatic sequence may be interrupted at any time to initiate a specific operational mode by depressing the appropriate Master Control Button. Once a manual mode selection is made, it remains until another mode selection is initiated. Once the automatic mode is reselected, the system continues to cycle as previously described. Should a



Figure 5-2. Photograph of the Central Processor and Control Unit (CPCU) of the TSC Airport Visibility System (ARVIS)



malfunction occur in the monitored circuits of the transmissometer, transmissometer receiver, projector lamp or projector power supply, a failure signal will be transmitted to the Master Control and a light indicator and an alarm signal will be triggered. The alarm may be turned off if the CPCU operator depresses the "failure acknowledge" button; however, the specific failure indicator will remain lighted until corrective field maintenance is implemented. The system could continue to operate but with the possibility of system performance degradation or damage.

The Photometric Display contains LED readouts arranged in columnar fashion. The second column, top to bottom, displays atmospheric transmittances for transmissometers at the touchdown, midpoint and rollout locations on the runway. These values are processed in the SDCU and displayed in the first column as RVR values at touchdown, midpoint and rollout. The third columnar display indicates the instrumented background luminance level on the runway ("automatic" switch setting) or alternatively, a value set in manually by the operator (i.e., bright, day, twilight or night switch setting). The fourth columnar display contains a LED readout indicating the specific runway monitored. A set of push-button switches is available for the insertion of HIRL settings 5, 4, 3, or 0 into the processor for RVR computations. In the automatic position, appropriate HIRL settings are fed automatically to the SDCU for RVR computation.

In the Data Recorders section of the CPCU there is a strip chart recorder and an incremental digital tape recorder. The strip chart recorder allows continuous atmospheric transmittance recording within .2 percent of full scale for any one of the transmissometers on the runway, selectable by means of its associated switch. Of greater significance, however, is the incorporation of a dual cassette incremental digital tape recorder which records all the available photometric data and ARVIS status. The information which is incrementally recorded every 10 seconds consists of the following: a) time in month, day, hour, minute, and second; b) runway light status, i.e., approach lights, sequenced flashing lights, background luminance input mode, background luminance; c) RVR for each of the

transmissometers; d) atmospheric transmittance for each of the three transmissometers; and e) failure mode status for all three transmissometers.

A software and hardware interface is supplied to read the information on the cassette and write the information on a teletype command which activates the cassette with the computer program. The program is read from the cassette and loaded in the SDCU. The cassette recorder provides historical evidence of total system conditions at all times to facilitate critical reviews of operational integrity, especially in accident investigation.

The Power Supply Accomplishes power control, conversion and distribution to the aforementioned in the CPCU.

#### 5.4 SIGNAL DATA CONVERTER UNIT

An experimental SDCU was developed by the Charles Stark Draper Laboratory, Inc. (Cambridge, Massachusetts) to satisfy the MOD II configuration with the capability for expansion to a MOD III. The detailed description of the SDCU is given in Appendix I.

## REFERENCES

1. Madden, P., M. Desai, and G. Mamom, "Study, Design and Implementation of a Signal Data Converter and Signal Simulator for a Visibility Measuring System," Final Report, R-806, July 1973, The Charles Stark Draper Laboratory, Inc., Cambridge MA 02139. (Reference appears as Appendix I in the report)
2. Ingrao, H.C. and J.R. Lifnitz, "Characteristics of a Signal Data Converter for a Multi-Runway Visibility Measuring System," Report No. DOT-TSC-FAA-72-1, October 1971.
3. "Preliminary Instruction Book - Runway Visual Range (RVR) System," Vol. I, July 1967, Prepared for the FAA contract FA6WA-1827 by Solid Radiation Inc., Los Angeles CA.
4. Ingrao, H.C. and J.R. Lifnitz, "Proposed Control Tower and Cockpit Visibility Readouts Based on an Airport-Aircraft Information Flow System," Report NO. DOT-TSC-FAA-71-18, July 1971.



## APPENDIX I

### STUDY, DESIGN AND IMPLEMENTATION OF A SIGNAL DATA CONVERTER AND SIGNAL SIMULATOR FOR A VISIBILITY MEASURING SYSTEM

This appendix was prepared by The Charles Stark Draper Laboratory, Inc. under Contract DOT-TSC-460 with the U. S. Department of Transportation, Transportation Systems Center, Optical Devices Group.

Publication of this appendix does not constitute approval by the U. S. Department of Transportation of the findings or conclusions contained herein. It is published for the exchange and stimulation of ideas.

TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1	INTRODUCTION. . . . .	I-1
2	DETERMINATION OF RUNWAY VISUAL RANGE (RVR). . . . .	I-4
	2.1 Introduction. . . . .	I-4
	2.2 Computation of RVR. . . . .	I-4
	2.3 Iterative Algorithms for Allard's Law . . . . .	I-6
	2.4 RVR Computation Accuracy. . . . .	I-14
	2.5 Fixed-Point Computation of RVR. . . . .	I-16
	2.6 Summary and Conclusions . . . . .	I-25
3	VISIBILITY SYSTEM SOFTWARE. . . . .	I-27
	3.1 Introduction. . . . .	I-27
	3.2 Basic Software Package. . . . .	I-27
	3.3 VISIB . . . . .	I-29
	3.4 LKSERV. . . . .	I-42
	3.5 EXECUT. . . . .	I-45
	3.6 Initialization. . . . .	I-59
4	MINICOMPUTER CONFIGURATION. . . . .	I-61
	4.1 Introduction. . . . .	I-61
	4.2 Simulator-Interface Minicomputer Configuration. . . . .	I-61
	4.3 Minicomputer Specification. . . . .	I-63
5	INTERFACE CONFIGURATION AND DESIGN. . . . .	I-64
	5.1 Introduction. . . . .	I-64
	5.2 Interface Configuration . . . . .	I-64
	5.3 Modular Interface Design. . . . .	I-67
	5.4 RVR Simulator-Interface Design. . . . .	I-68
	5.5 Implementation of the Simulator-Interface . . . . .	I-78

TABLE OF CONTENTS (Cont.)

<u>Section</u>	<u>Page</u>
5.6 Characterization of the Interface Sensor Inputs . . .	I-79
5.7 Interface Specification . . . . .	I-82
 <u>Appendix</u>	
A ITERATIVE ALGORITHMS FOR RVR COMPUTATION. . . . .	I-83
A.1 Introduction. . . . .	I-83
A.2 Newton-Raphson Method . . . . .	I-83
A.3 Higher-Order Methods Utilizing Second Derivative. . .	I-84
A.4 Method 2. . . . .	I-85
A.5 Halley's Method . . . . .	I-85
A.6 Modified Newton-Raphson Method. . . . .	I-87
A.7 Comments on the Order of Convergence. . . . .	I-87
 B COMPUTATION OF LOGARITHM. . . . .	 I-88
 C ATMOSPHERIC TRANSMITTANCE FOR RVR RANGE 50 TO 5900 FEET . .	 I-90
 D SIMULATOR SOFTWARE. . . . .	 I-107
D.1 Software Description. . . . .	I-107
D.2 Typical Teletype I/O. . . . .	I-110
D.3 PDP-11 Instruction Set. . . . .	I-111
D.4 RVR Update Time Requirements. . . . .	I-113
D.5 Program Size. . . . .	I-113
D.6 Simulator Software. . . . .	I-115
 REFERENCES. . . . .	 I-167

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
2-1	Plot of $f(V)$ . . . . . I-7
2-2	Truncation error in RVR due to limited wordlength in fixed-point arithmetic calculations of RVR using Allard's law for four different values of atmospheric transmittances $t_b$ (and associated correct RVR) with $E_t = 2$ mile-candles, $I = 10,000$ candles, and baselength $b = 60$ feet. . . . . I-24
2-3	Plot of $\Delta V(\%)$ vs $t_b(\%)$ for nighttime conditions showing the dependence of truncation error in the calculation of $V$ on the atmospheric transmittance $t_b$ for four different word-bit lengths. . . . . I-25
2-4	Plot of $\Delta V(\%)$ vs $t_b(\%)$ for daytime conditions showing the dependence of truncation error in the calculation of $V$ on the atmospheric transmittance $t_b$ for four different word-bit lengths. . . . . I-26
3-1	Primary modules comprising an airport visibility system software developed by the Draper Laboratory. . . . . I-28
3-2	Basic airport visibility system package which handles during an update cycle: data inputs, calculations, transmission, recording display, and control functions associated with the system. . . . . I-30
3-3	Routine to input data . . . . . I-31
3-4	Transmissometer data input routine. . . . . I-32
3-5	Timing relationships for different software modules with respect to the transmissometer pulse counting during an update cycle. . . . . I-34
3-6	Runway-light-intensity input routine. . . . . I-35
3-7	Sky-background-luminance input routine. . . . . I-35
3-8	Routine to read transmissometer selector switch position for RVR display. . . . . I-36
3-9	Routine to read inputs from SVR interface . . . . . I-36
3-10	Routine to read TVR inputs. . . . . I-37
3-11	Routine to read ceilometer inputs . . . . . I-37
3-12	Airport visibility system parameters calculation routine. . I-38
3-13	Routine to compute RVR. . . . . I-39



LIST OF ILLUSTRATIONS (Cont.)

<u>Figure</u>		<u>Page</u>
3-14	Iteration routine to compute RVR. . . . .	I-40
3-15	Routine to compute SVR. . . . .	I-41
3-16	Routine to compute TVR. . . . .	I-41
3-17	Routine to compute ceiling. . . . .	I-42
3-18	Output routine. . . . .	I-43
3-19	Routine to service air-traffic-controller display . . . . .	I-44
3-20	Routine to service air-traffic-control display. . . . .	I-46
3-21	Routine to display RVR. . . . .	I-46
3-22	Routine to display SVR. . . . .	I-47
3-23	Routine to display TVR. . . . .	I-47
3-24	Routine to display ceiling. . . . .	I-47
3-25	Routine for system data recording . . . . .	I-48
3-26	Routine for information transmission system . . . . .	I-48
3-27	Light control routine . . . . .	I-48
3-28	Approach light control routine. . . . .	I-49
3-29	Runway light intensity control routine. . . . .	I-49
3-30	Routine to keep time and service time-related control functions . . . . .	I-50
3-31	Executive routine . . . . .	I-51
3-32	Routine to enter basic system parameters. . . . .	I-53
3-33	Routine for transfer to normal operating mode . . . . .	I-54
3-34	Test routine to check out VISIB . . . . .	I-54
3-35	Routine to determine number of days in a month. . . . .	I-55
3-36	Routine to set basic system parameters. . . . .	I-56
3-37	Routine to input RVR system parameters. . . . .	I-57
3-38	Routine to input SVR system parameters. . . . .	I-58
3-39	Routine to input TVR system parameters. . . . .	I-58
3-40	Routine to input ceilometer parameters. . . . .	I-59
3-41	Starting routine. . . . .	I-59
3-42	Initialization routine. . . . .	I-60
5-1	Typical bus line. . . . .	I-65
5-2	Typical signal data converter I/O interface configuration . . . . .	I-66
5-3	Block diagram of the simulator-interface. . . . .	I-69
5-4	Detailed schematic of the simulator-interface . . . . .	I-70

LIST OF ILLUSTRATIONS (Cont.)

<u>Figure</u>		<u>Page</u>
5-5	Front control panel of the PDP-11/10-AC mini-computer of the simulator-interface minicomputer configuration. . . . .	I-75
5-6	ASR-33 teletype and slow-speed paper-tape reader/punch. . . . .	I-75
5-7	Front control panel of the signal simulator bearing switches controlling the simulated sensor inputs to the minicomputer and the visual RVR digital display. . . . .	I-75
B-1	Plot of $-\log_2 \epsilon$ vs $N$ , showing the dependence of $\epsilon$ , the computation error for the fixed-point algorithm and $N$ , bit length of the computer word . . . . .	I-89

LIST OF TABLES

<u>Table</u>	<u>Page</u>
2-1	Computer printout showing convergence for different algorithms to compute RVR. . . . . I-11
2-2	Predicted and actual error in RVR, due to scaling in computation of RVR from Allard's law, using fixed-point calculations, for four different transmittances ( $E_t = 2$ mile-candles, $I = 10,000$ candles, $b = 60$ feet). . . I-19
2-3	Range of quantities involved in calculating RVR from Allard's law using Newton-Raphson method. . . . . I-21
3-1	EXECUT commands and responses . . . . . I-52
5-1	Pulse-train discrete-frequency code . . . . . I-76
5-2	Typical sequence of I/O operations for the simulator-interface . . . . . I-78
C-1	Atmospheric transmittance . . . . . I-91
D-1	Typical teletype input/output printout. . . . . I-110
D-2	Estimates of processor time requirements for RVR update . . I-113
D-3	Size of the program simulator software. . . . . I-114

SYMBOL DEFINITION AND UNITS

<u>Symbol</u>	<u>Definition</u>	<u>Unit</u>
b	Baselength of transmissometer	feet
B	Background luminance	candles/ft. <sup>-2</sup>
E <sub>t</sub>	Visual illuminance threshold	mile-candles
f(V)	Functional form of Allard's Law	
f <sub>v</sub> , f <sub>k</sub>	f(V) = f <sub>k</sub> = f <sub>v</sub> , Intensity of runway lights	
I	Intensity of runway lights	candles
N	Number of bits in computer word	
N	Number of transmissometers feeding an SDCU	
N-R	Newton-Raphson method	
t <sub>b</sub>	Atmospheric transmittance over baselength	
V	RVR, runway visual range	feet
V <sub>a</sub>	RVR per Allard's Law	feet
V <sub>k</sub>	RVR per Koschmeider's Law	feet
2 <sup>S</sup>	Scale factor	
β <sub>1</sub>	Multiplying factor for Newton-Raphson method	
ε	Visual contrast threshold	
ε	Average error of the logarithm function	
α	Multiplying factor	

## 1. INTRODUCTION

Visibility information at U.S. airports relies upon indirect measurements based on atmospheric transmittance sensors (transmissometers) and its associated dedicated hardwired-logic controller-computer. The planned evolution<sup>(1,2)</sup> of a comprehensive integrated airport runway visibility measuring system envisages a multi-sensor system requiring a greatly expanded data acquisition, computation, logging, and display capability compared to the current FAA system.

There are many compelling motivations for using a programmable minicomputer instead of special-purpose hardwired logic. One of these motivations is the flexibility afforded by programmable systems. The software system contains the details of the application which are subject to substantial change during evolutionary development, and even from one operational facility to another. Such software systems also permit the writing of diagnostic programs that can be used for the efficient debugging of the hardware system. Because of the interdependence between the hardware and software portions of a system, the system offers a high potential for growth, and associated with this, a much longer useful life. Most real-time applications of this type are dynamic in nature. Each successful application often reveals other associated applications that are economically, technically, or administratively desirable. Programmable systems provide the flexibility to economically accommodate these dynamic system requirements.

Despite their flexibility, contemporary programmable systems often represent the lowest cost implementation. This is probably the most important motivation of all for using minicomputers. The use of general-purpose integrated components made in large quantities is the main reason for this cost advantage. Mini- and microcomputer (or microprocessor) technology is currently extremely dynamic, primarily because of advances in semiconductor component manufacturing technology. It has been predicted that entire processors will be fabricated on one

chip at a cost of perhaps \$10 to \$20. Present-day cost is hardly much higher. This is completely within the realm of possibility today. The trend is toward increasing capability and lower costs for the main frame, central processing units, and I/O structures, and further improvement in the price/performance ratio of the future computer peripherals.

The following sections describe an investigation into some aspects important to the implementation of a minicomputer-based airport visibility measuring and control, data-logging and display system. The tasks included the following:

- (1) Derivation of efficient algorithms for RVR.
- (2) Accuracy versus wordlength trade-off studies.
- (3) Conceptual design of software for the planned evolutionary visibility measuring systems.
- (4) Design and construction of an experimental minicomputer based visibility-measuring Signal Data Converter Unit (SDCU) with simulated sensor inputs.
- (5) Design and implementation of software for the experimental systems.
- (6) Characterization of computer input and output data.
- (7) Specification of a minicomputer portion of the SDCU and associated interface to implement the planned visibility system.

The design and construction of the experimental simulator-interface and associated software was useful in identifying special problems and served as a guide to the conceptual design of software and the specification of hardware for the planned visibility system. In addition, it demonstrates the flexibility and scope of a minicomputer-based SDCU in this application. The software developed for the experimental system may be thought of as the prototype of that required for the planned evolutionary system; it contains all submodules required for the eventual systems.

Software design and accuracy trade-off studies are extremely important tasks in the subject application. Another important task is the question of a cost-effective way of providing sufficient redundancy in the system to facilitate uninterrupted service by various measures such as the provision of complementary items, detection and

appropriate action on the occurrence of anomalous and faulty system operation conditions, etc. Efficient software and thorough trade-off studies lead to a cost-effective specification for the minicomputer configuration and its associated interface. Substantial investigative effort in the above areas is justified because it is a one-time task, while multiple hardware units will be deployed when the eventual operational system is implemented.

## 2. DETERMINATION OF RUNWAY VISUAL RANGE (RVR)

### 2.1 INTRODUCTION

RVR is the maximum distance along the runway in the direction of take-off or landing at which the runway, or the specified lights, or markers delineating it, can be seen from a position above a specified point on its center line at a height corresponding to the average eye level of the pilot at touchdown. The calculation of RVR is based on the measurements of the atmospheric transmittance over a specified baselength by transmissometers located along the runway. The value of RVR that is reported is the higher value of RVR based on either the sighting of high-intensity runway lights or the sighting of the runway markers by contrast.

In this section, various aspects related to the calculation of RVR using fixed-point limited-wordlength arithmetic are considered. Allard's and Koschmieder's laws used in the calculation are described in Subsection 2.2. Algorithms for the calculation of RVR and the accuracy characteristics of the RVR calculations are considered in Subsection 2.3 and 2.4 respectively. The effect of limited-wordlength fixed-point arithmetic on the accuracy of RVR determination is considered in Subsection 2.5. Conclusions are presented in Subsection 2.6.

### 2.2 COMPUTATION OF RVR

The computation of RVR is based on either of the relations:

$$\epsilon = (t_b)^{V/b}, \quad (\text{Koschmieder's Law}) \quad (1)$$

or

$$E_t = \frac{I}{(V/5280)^2} \cdot (t_b)^{V/b}, \quad (\text{Allard's Law}) \quad (2)$$

where

$V$  = RVR, in feet.

$b$  = baselength of transmissometer, in feet.



- $t_b$  = atmospheric transmittance over baselength  $b$  ( $0 \leq t_b \leq 1$ ).  
 $\epsilon$  = visual-contrast threshold.  
 $E_t$  = visual-illuminance threshold of pilot, in mile-candles.  
 $I$  = intensity of runway lights, in candles.

Equation (1) represents Koschmieder's law, and is based on using contrast between an unlighted object and its background as a criterion for its visibility. Usually a contrast threshold of 5% is used for RVR calculations. Equation (2) is Allard's law for visibility of lighted sources such as runway lights. Unlike Koschmieder's law, the illuminance threshold is not constant, and depends on the background luminance,  $B$ . An empirical relationship between the two has been established by Blind Landing Experimental Unit (BLEU) and is given by

$$\log E_t = 1.3733 + 0.64 \log B \quad (3)$$

where

$$B = \text{background luminance, in candles/ft}^2.$$

During good daytime visibility, the runway or its markings are more readily sensed by the pilot than runway lights, whereas the reverse is true during nighttime and most daytime fogs. This is reflected in the value of RVR obtained using either of the laws of Eq. (1) and Eq. (2). Thus, for RVR reporting, the higher of the two computed values is chosen.

Logarithmic transformation of Eq. (1) and Eq. (2) yields the following computationally convenient forms:

$$V = \frac{b \log \epsilon}{\log t_b} \quad (\text{Koschmieder's Law}) \quad (4)$$

and

$$f(V, t_b, I, E_t, b) = 0 \quad (\text{Allard's Law}) \quad (5)$$

where

$$f(V, t_b, I, E_t, b) = \frac{1}{2} \log E_t - \frac{1}{2} \log I - \log 5280 + \log V - \frac{V}{2b} \log t_b$$

RVR for a given  $t_b$ , over a baselength  $b$ ,  $B$ , and  $I$  can be obtained by choosing the higher RVR solution of Eq. (4) and Eq. (5). Koschmieder RVR can be evaluated explicitly in terms of  $b$ ,  $\epsilon$ , and  $t_b$  using Eq. (4).

On the other hand, the evaluation of Allard's RVR involves solving Eq. (5) which is a nonlinear implicit functional relation between  $V$  and data inputs  $b$ ,  $t_b$ ,  $E_t$ , and  $I$ . A number of iterative schemes, of varying degrees of complexity are available in the literature to solve nonlinear equations of the type  $f = 0$ . The following subsection considers some of these algorithms in the context of Eq. (5). In Subsection 2.3, we shall primarily restrict our consideration to finding efficient algorithms for computing the solution of Eq. (5).

## 2.3 ITERATIVE ALGORITHMS FOR ALLARD'S LAW

### 2.3.1 Iterative Algorithms

Figure 2-1 shows plots of  $f(V)$  for four different values of  $t_b$  and a given set of values for  $b$ ,  $E_t$ , and  $I$ . The first and the second derivatives of  $f(V)$  are:

$$f'(V) = \frac{1}{V} - \frac{1}{2b} \log t_b, \quad (6)$$

$$f''(V) = -\frac{1}{V^2}. \quad (7)$$

The function  $f(V)$  is a well-behaved function of  $V$ ,\* possessing a unique zero, and its slope and higher derivatives do not change their sign with variation in  $V$ . These features are important in the consideration of iterative schemes to find the zero.

We restrict our consideration to single-point algorithms of the type

$$V_{i+1} = V_i - \alpha_i f(V_i) \quad (8)$$

which utilize the information from  $f(V)$  and its higher derivatives at a given point.  $\alpha_i$  is a multiplying factor to be suitably chosen. With such algorithms, it may be easily ascertained that convergence to the zero is assured for any initial guess of  $V$ , as long as both  $V_i$  and  $\alpha_i$  are greater than zero. On the other hand, convergence is not readily assured in case of multi-point algorithms, which utilize information about the function at more than one point.

---

\*Except near the singularity at  $V = 0$ , where the value of  $f(V)$  tends to minus infinity. By posting a lower limit on the range of variation of  $V$ , the singularity and the attendant problems are easily avoided in numerical computations. The limit can be as low as 1.

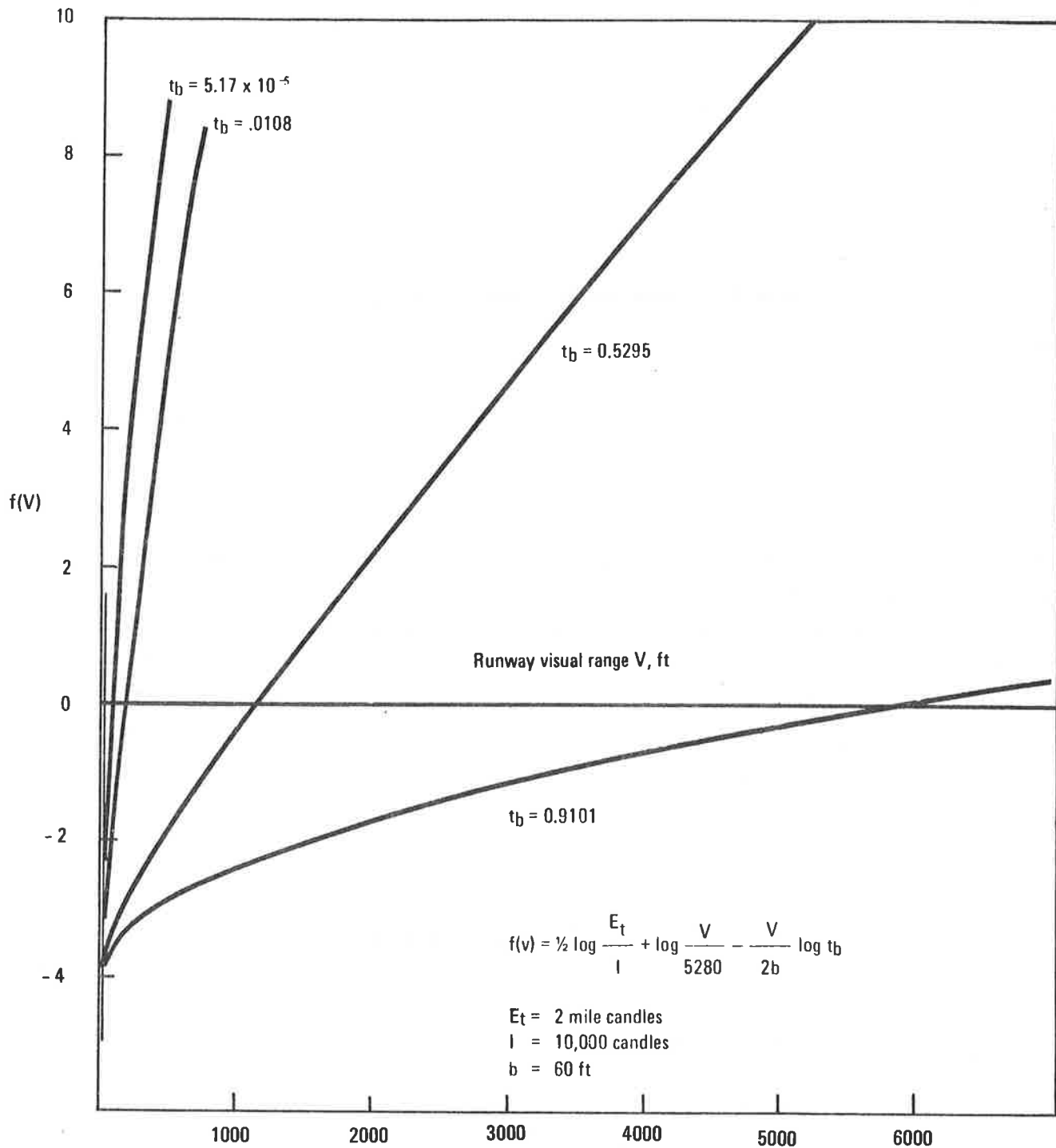


Figure 2-1. Plot of  $f(V)$ . Allard's law satisfies when  $f(V) = 0$ . By posting a lower limit on the range of variation of  $V$ , the singularity and the attendant problems are easily avoided in numerical computations. The limit can be as low as 1.

Three single-point iterative algorithms considered here for the solution of Eq. (5) are:\*

$$V_{i+1} = V_i - f(V_i)/f'(V_i) \quad (9)$$

Newton-Raphson Method

$$V_{i+1} = V_i - u(V_i)/u'(V_i) \quad (10)$$

$$u(V_i) = f(V_i)/f'(V_i),$$

Modified Newton-Raphson Method,

and

$$V_{i+1} = V_i - u(V_i) \cdot \frac{1}{1 - \frac{1/2 u(V_i) f''(V_i)}{f'(V_i)}} \quad (11)$$

Halley's Method.

The estimate of zero of  $f(V)$  is improved by utilizing the information about the value of the function and its slope at the present estimate in the case of Newton-Raphson (N-R) method and also additionally utilizing the information about the curvature in the case of the other two higher-order methods.

Substituting for the values of  $f'(V_i)$ ,  $f''(V_i)$ ,  $u(V_i)$ , and  $u'(V_i)$  in Eq. (9) to (11), we get:

$$V_{i+1} = V_i - \alpha_i f(V_i), \quad (12)$$

$$\alpha_i = \beta_i \cdot \frac{V_i}{\left(1 - \frac{V_i}{2b} \log t_b\right)}$$

where

$$\beta_i = 1 \quad \text{for N-R method,} \quad (13)$$

$$\beta_i = \frac{1}{1 + \frac{f}{\left(1 - \frac{V_i}{2b} \log t_b\right)^2}} \quad \text{for modified N-R method,} \quad (14)$$

\* Further details on the derivation and the performance of these algorithms are given in Appendix A.

$$\beta_i = \frac{1}{1 + 1/2 \frac{f}{\left(1 - \frac{V_i}{2b} \log t_b\right)^2}} \quad \text{for Halley's method.} \quad (15)$$

The multiplying factor  $\beta$  for modified N-R and Halley's methods takes into account the curvature of the function.

For assured convergence, both  $\alpha_i$  and  $V_i$  should be greater than zero. Further restrictions on the magnitude of  $\beta_i$  and  $V_i$  need to be imposed as shown below to meet the above requirements.

$$V_i > 0 \quad \text{for N-R method} \quad (16)$$

$$\left. \begin{array}{l} V_i > 0, \\ \beta_i > \gamma > 0 \end{array} \right\} \text{for modified N-R method} \quad (17)$$

$$\left. \begin{array}{l} V_i > 0, \\ \beta_i > \gamma > 0 \end{array} \right\} \text{for Halley's method} \quad (18)$$

The limit  $\gamma$  is to be suitably chosen in Eq. (17) and Eq. (18). A further restriction on the maximum value of  $V_i$  may be required to prevent overflow during the fixed-point computations.

### 2.3.2 Selection of an Algorithm

The order of convergence (i.e., how fast the convergence is) could form the basis of choice for the selection of an iterative algorithm. However, it is more realistic and meaningful to compare the computational efficiency, which measures the total amount of numerical computations that need to be done to arrive at a given accuracy in finding the zero. The total amount of computations depends upon the initial estimate, numerical computations (sometimes termed as cost) per iteration and total number of iterations necessary to arrive at a given accuracy in finding the zero. The numerical costs associated with the three algorithms are of the same order, since the increase in the computations involved with modified N-R and Halley's methods is very small. The total number of iterations required depends on the initial estimate, the order of accuracy required, the order of convergence of the algorithm, and the nature of  $f(V)$ . It may seem from Figure 2-1 and Eq. (7) that  $f(V)$  is nearly linear except at low values of  $V$ . Thus, it may be expected that the performance of the three algorithms would be similar.

Table 2-1 shows the results of a convergence study of different algorithms carried out on a digital computer.\* RVR is computed for eight different values of transmittance,  $t_b$ , over a baselength of 60 feet.  $E_t = 2$  mile-candles and runway high-intensity light of 10,000 candles. The starting guess for all iterative schemes is 1000 feet. Table 2-1 lists the values of RVR  $V(I)$  at different stages of iteration,  $I$ , for four different algorithms. The table includes Method 2, which is a variation of the two higher-order methods described in Subsection 2.3.1. Appendix A includes the description, including the derivation of Method 2. The exit criterion employed for this study is given by  $|\Delta V_i| \leq 10^{-4}$ . For the modified N-R method and Halley's method,  $\gamma = 0.5$  was employed. The following observations are made with respect to the above study:

- (1) For low-transmittance values (low RVR) the performance of all four methods is nearly similar, with that of the N-R method slightly better than those of the other three methods.
- (2) For high transmittance values (high RVR) the three methods using the second derivative perform slightly better than the N-R method. As expected, for initial guesses far off from the actual RVR, the next iterates are much closer to the actual RVR for these methods than for the N-R method. The performance of the modified N-R method is slightly superior to the other three methods.
- (3) The selection from amongst the four methods would primarily depend upon the level of accuracy specified. For example, for the high accuracy of the order of  $10^{-4}$  used for study, the number of iterations needed to reach this high accuracy are roughly the same for all methods. However, for low accuracy requirements, the number of iterations for the three methods using the second derivative are slightly less than that for N-R method. However, for these higher-order methods, this slight advantage in the total number of iterations is offset by the small increase in the numerical cost per iteration.

One of the conclusions that emerges from the above observations is that there is not much to be gained by way of convergence and computational efficiency in going from the simpler N-R method to the higher-order methods such as modified N-R or Halley's methods.

---

\*Because of the nature of the study, floating-point arithmetic was used.

TABLE 2-1. COMPUTER PRINTOUT SHOWING CONVERGENCE FOR DIFFERENT ALGORITHMS TO COMPUTE RVR

ET= 2.000000  
 INT=20000.000000  
 BASE LENGTH= 60.000000

TS= .012860

I	V(I)			
	N-R	NO. 2	M0D. N-R	HALLEY
1	222.162907	240.216702	256.668720	239.807174
2	216.016616	218.196882	220.516118	218.147888
3	215.257133	215.535975	215.840268	215.529743
4	215.159844	215.195734	215.235044	215.194932
5	215.147308	215.151932	215.157000	215.151829
6	.000000	215.146289	215.146942	215.146275

TS= .213200

I	V(I)			
	N-R	NO. 2	M0D. N-R	HALLEY
1	564.596618	578.972733	591.567784	578.513235
2	539.185700	541.228057	543.312205	541.165753
3	535.828154	536.117468	536.419772	536.108730
4	535.357059	535.398037	535.441002	535.396801
5	535.290423	535.296227	535.302315	535.296052
6	535.280986	535.281808	535.282671	535.281784

TS= .398990

I	V(I)			
	N-R	NO. 2	M0D. N-R	HALLEY
1	860.661609	862.905975	865.010251	862.870397
2	844.132236	844.499905	844.865831	844.494595
3	841.741202	841.796673	841.852384	841.795885
4	841.386556	841.394835	841.403160	841.394717
5	841.333761	841.334995	841.336236	841.334977

TS= .483940

I	V(I)			
	N-R	NO. 2	M0D. N-R	HALLEY
1	1026.441559	1026.537949	1026.635754	1026.538301
2	1030.636902	1030.653989	1030.671118	1030.654045
3	1031.280717	1031.283379	1031.286042	1031.283387
4	1031.379000	1031.379407	1031.379815	1031.379409

TABLE 2-1. COMPUTER PRINTOUT SHOWING CONVERGENCE FOR DIFFERENT ALGORITHMS TO COMPUTE RVR (Cont.)

I3 = .647300

I	V(I)			
	N-R	N0. 2	M0D. N-R	HALLEY
1	1446.177337	1484.845741	1539.729774	1488.514974
2	1572.393895	1530.804513	1590.341875	1581.443206
3	1595.440270	1596.860675	1598.413128	1596.964552
4	1599.229274	1599.469349	1599.709446	1599.476081
5	1599.840572	1599.827610	1599.917833	1599.880300
6	1599.938913	1599.944869	1599.951336	1599.945301

I4 = .723100

I	V(I)			
	N-R	N0. 2	M0D. N-R	HALLEY
1	1736.756079	1863.377142	2122.634657	1889.654758
2	2075.809530	2057.978717	2103.270132	2062.844035
3	2086.682716	2092.443683	2100.043380	2093.267605
4	2097.269739	2098.240330	2099.506426	2098.377803
5	2099.044613	2099.206420	2099.417095	2099.229301
6	2099.340258	2099.267185	.000000	2099.370992
7	2099.339452	.000000	.000000	.000000

I5 = .912600

I	V(I)			
	N-R	N0. 2	M0D. N-R	HALLEY
1	2796.878096	3695.317145	4593.756193	4593.756193
2	4757.852330	5395.261349	5689.248529	5621.516235
3	5627.057781	5796.113044	5853.836383	5843.282148
4	5842.924289	5877.747026	5888.972402	5886.898316
5	5886.774212	5893.541591	5895.698836	5895.299475
6	5895.273706	5896.523993	5896.987621	5896.911020
7	5896.976103	5897.155316	5897.234590	5897.219908
8	5897.211944	.000000	.000000	.000000

I6 = .836500

I	V(I)			
	N-R	N0. 2	M0D. N-R	HALLEY
1	2263.217516	2748.345159	3526.495632	3050.753806
2	3099.308563	3272.191304	3401.192166	3333.785640
3	3340.013001	3275.637115	3402.448929	3386.848792
4	3387.852456	3394.357721	3399.123328	3396.354624
5	3396.531411	3397.688557	3398.533723	3398.042831
6	3398.074076	3398.279377	3398.429206	3398.342186
7	3398.347722	3398.384119	.000000	.000000



### 2.3.3 Steps for an Algorithm to Compute RVR

An iterative algorithm based on Eq. (12) to (18) is described in the following. The algorithm computes RVR based on  $E_t$ ,  $I$ , and  $t_b$ .

The algorithm consists of the following steps:

- (1) For data inputs  $E_t$ ,  $I$ , and  $t_b$ , calculate  $\log E_t$ ,  $\log I$ ,  $\log t_b$ , and  $f_k$  where  $f_k = 1/2(\log E_t - \log I) - \log 5280$ .
- (2) Consider an initial guess  $V_0$  for RVR which may be based upon the previous or old RVR calculated, or may be a fixed quantity.
- (3) Calculate

$$f_v = \log V_i - \frac{V_i \log t_b}{2b}$$

and

$$f(V_i) = f_k + f_v.$$

- (4) Calculate

$$\Delta V_i = \beta_i \frac{V_i f(V_i)}{1 - \frac{V_i}{2b} \log t_b}$$

with

$$\begin{aligned} \beta_i &= 1 && \text{for N-R method,} \\ \beta_i &= \frac{1}{1 + X} && \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{for modified N-R method,} \\ X &\geq \gamma - 1 \\ \beta_i &= \frac{1}{1 + \frac{X}{2}} && \left. \begin{array}{l} \\ \\ \end{array} \right\} \text{for Halley's method,} \\ \frac{X}{2} &\geq \gamma - 1 \end{aligned}$$

where

$$X = \frac{f(V_i)}{\left(1 - \frac{V_i}{2b} \log t_b\right)^2}$$

and  $\gamma$  is a given positive number.

(5) Calculate

$$V_{i+1} = V_i - \Delta V_i,$$

subject to the restrictions,

$$V_{\min} \leq V_{i+1} \leq V_{\max}$$

where  $V_{\min}$  and  $V_{\max}$  are given.

(6) Check if a given exit\* criterion is satisfied. If not, go to Step (3).

(7) The RVR so calculated is Allard's RVR, say  $V_A$ . Calculate  $V_K$ , Koschmieder RVR from

$$V_K = \frac{b \log \epsilon}{\log t_b},$$

and where

$$\epsilon = 0.05.$$

For RVR reporting, choose the higher of the two values,  $V_A$  and  $V_K$ .

## 2.4 RVR COMPUTATION ACCURACY

Accuracy characteristics of RVR computations are considered in Subsections 2.4.1 through 2.4.3. Error in the determination of RVR may arise due to error measurement of input parameters  $E_t$ ,  $I$ , and  $t_b$ , and due to truncation errors in the arithmetic involved in the calculation of RVR from Allard's law and Koschmieder's law.

### 2.4.1 Allard's Law

Let  $V$  represent the RVR for a set of input data  $E_t$ ,  $I$ , and  $t_b$ . Consider a change in data inputs to  $E_t + dE_t$ ,  $I + dI$ , and  $t_b + dt_b$ . Corresponding change  $dV$ , in the value of RVR, is given by

$$f(V + dV, E_t + dE_t, I + dI, t_b + dt_b) - f(V, E_t, I, t_b) = 0$$

or

\* Since exact RVR is not known, the exit criterion may be based on how close to zero either  $f(V_i)$  or  $\Delta V_i$  is satisfied.

$$\begin{aligned} \frac{1}{2} \log \left( 1 + \frac{dE_t}{E_t} \right) - \frac{1}{2} \log \left( 1 + \frac{dI}{I} \right) + \log \left( 1 + \frac{dV}{V} \right) \\ - \frac{V}{2b} \log \left( 1 + \frac{dt_b}{t_b} \right) - \frac{dV}{2b} \log \left( t_b + dt_b \right) = 0. \end{aligned}$$

The change in RVR can be explicitly evaluated from the above expression if the percentage changes (or errors) in the nominal values of  $E_t$ ,  $I$ , and  $t_b$  are small. Thus, we get:

$$\frac{dV}{V} = \frac{2 \frac{1}{V} \left( \frac{dI}{I} - \frac{dE_t}{E_t} \right) + \frac{1}{b} \cdot \frac{dt_b}{t_b}}{f'(V)} \quad (19)$$

where

$$f'(V) = \frac{1}{V} - \frac{1}{2b} \log t_b$$

Equation (19) relates the effect of errors in  $I$ ,  $E_t$ , and  $t_b$  on the determination of RVR from Allard's law. The following observations may be made with respect to Eq. (19).

- (1) The percentage error in RVR is inversely proportional to  $f'(V)$ , the slope of  $f(V)$  at its zero. It may be seen from Figure 2-1 that the slope is larger at higher values of RVR. Thus, a given percentage in the input data,  $E_t$ ,  $I$ , and  $t_b$  leads to a larger percentage change in RVR during high RVR conditions and a smaller change during low RVR conditions.
- (2) Larger percentage errors can be tolerated for  $I$  or  $E_t$  than for  $t_b$  because of the weighting factors  $V$  and  $b$  involved in each case. The effect of errors in  $I$  or  $E_t$  becomes unimportant during low-visibility conditions.
- (3) For large  $t_b$  (or large  $V$ ),

$$\frac{dV}{V} \approx - \frac{d(\log t_b)}{\log t_b} \quad (20)$$

#### 2.4.2 Koschmieder's Law

By differentiating Eq. (3) we obtain the following relationship between  $dV$  and  $dt_b$ .

$$\frac{dV}{V} = - \frac{d(\log t_b)}{\log t_b} \quad (21)$$

It may be noted that the effect of errors in  $t_b$  is similar for both Koschmieder's and Allard's laws during good visibility conditions.

#### 2.4.3 Effect of Truncation Error

Error in the computation of a logarithm function is the major source of error, due to loss of significance in the calculation of RVR. Equations (19) and (21) also may be utilized to predict the effect of truncation error in the computation of the logarithm of  $V$  and the input values  $E_t$ ,  $I$ , and  $t_b$ . Thus, for Allard's law we get:

$$\frac{dV}{V} = \frac{\frac{1}{2V} [d(\log I) - d(\log E_t)] + \frac{1}{b} d(\log t_b)}{f'(V)}$$

The effect of wordlength and truncation on the accuracy of RVR determination is considered in Subsection 2.5.

#### 2.5 FIXED-POINT COMPUTATION OF RVR

In fixed-point calculations suitable scaling of numbers becomes necessary to avoid the overflow or loss of significance likely to result during the arithmetical operations on numbers whose magnitudes may vary over a wide range. The numbers can be scaled up or down by the use of a suitable multiplying factor chosen to satisfy the significance requirements needed in the arithmetical operations.

The most involved calculations in the computation of RVR arise in the calculation of logarithms of various quantities such as  $E_t$ ,  $I$ ,  $t_b$ , and  $V$ . It has been shown in Subsection 2.4 how the accuracy in the determination of RVR is affected by the loss of significance due to truncation in the calculation of logarithm of input quantities  $E_t$ ,  $I$ , and  $t_b$ . The loss of accuracy can be avoided or kept to a minimum provided unlimited wordlength is available for the fixed-point arithmetic and a suitable scaling is adopted. However, in case of limited-wordlength arithmetic, the magnitude of the scale factor and consequently the accuracy attainable are governed by the overflow problems associated with the limited wordlength.

The rest of the section is devoted to the consideration of:

- (1) Scaling and its effect on the accuracy of RVR determination.
- (2) Overflow problems associated with scaling and the measures to avoid overflow during RVR computations.
- (3) Order of accuracy attainable for a given wordlength.

A limited-wordlength integer arithmetic is assumed in the discussions that follow.

### 2.5.1 Scaling

Consider a scale factor of  $2^S$  associated with the computation of RVR laws. By multiplying Eq. (5) by  $2^S$  we have:

$$\begin{aligned} 2^S f(V) &= \frac{1}{2} 2^S \log E_t - \frac{1}{2} 2^S \log I - 2^S \log 5280 \\ &+ 2^S \log V - \frac{V}{2b} \cdot 2^S \log t_b \end{aligned} \quad (22)$$

The operations of addition, subtraction, multiplication, and division are carried out in single-precision-integer arithmetic in the evaluation of  $2^S f(V)$ .<sup>\*</sup> The evaluation of the logarithm of  $E_t$ ,  $I$ ,  $t_b$ , and  $V$  is further assumed to be handled by a software routine that employs an efficient high-accuracy algorithm employing integer arithmetic. Appendix B describes such an algorithm for calculating the logarithm of an integer. It is also shown that the order of accuracy associated with this algorithm is approximately  $2^{-(N-5)}$ , where  $N$  = length of the computer word that represents a single-precision number, including a bit for the sign. Thus, the error in the scaled-integer representation of a logarithm of a number, such as  $2^S \log I$ , is of the order of  $2^{-s}$  for  $s \leq N - 5$ , and represents the dominant source of error in the RVR calculations.

We shall consider the error introduced in the determination of RVR due to the truncation in the calculation of logarithm. Let  $V$  represent the correct RVR for a set of input data  $E_t$ ,  $I$ , and  $t_b$ . The change in the value of  $f(V)$  due to the error in the logarithm of this set of data is given by

---

\*An operation, such as  $I \cdot J/K$ , involving multiplication and division among single-precision numbers may be considered a single operation, provided the single-precision-integer arithmetic associated with the computer provides a double-precision product of  $I$  and  $J$ , which in turn is used as a double-precision dividend for the divide operation by  $K$ . Most of the computers have such a hardware or software capability associated with the single-precision multiply and divide operations.

$$2^S df = \frac{1}{2} \Delta (2^S \log E_t) - \frac{1}{2} \Delta (2^S \log I) - \frac{V}{2b} \Delta (2^S \log t_b), \quad (23)$$

where  $\Delta$  represents the truncation error.

If  $2^S df \geq 1$ , this leads to an error in the RVR determination. The error,  $dV$  in the value determined by RVR algorithms due to this error in  $df$  can be estimated to first order in  $dV$  from the relation:

$$dV \left( \frac{2^S}{2V} - \frac{2^S}{2b} \log t_b \right) = -2^S df \pm 1. \quad (24)$$

The right-hand side of Eq. (24) represents the change in  $f(V)$  due to  $dV$  necessary to cancel the effect of truncation error in the calculation of the logarithm of  $E_t$ ,  $I$ , and  $t_b$ . From Eq. (23) and Eq. (24) we obtain:

$$dV = \frac{1 + \frac{V}{2b} (2^S \log t_b) + \frac{1}{2} (2^S \log E_t) - \frac{1}{2} (2^S \log I)}{\frac{2^S}{V} - \frac{2^S}{2b} \log t_b} \quad (25)$$

Now, the magnitudes of  $2^S \log t_b$ ,  $2^S \log E_t$ , and  $2^S \log I$  are all less than 1. Thus, for  $V > 2b$ ,

$$dV \approx \frac{V}{2bf'(V)} \cdot \frac{1}{2^S} \quad (26)$$

Equation (26) represents the order of accuracy associated with fixed-point calculations for determining RVR from Allard's law. The order of accuracy is seen to depend upon the order of significance retained in the logarithmic representation of numbers and upon the value of the atmospheric transmittance, which in turn determines the slope of the function and the magnitude of RVR. For  $V \gg 2b$ , i.e., large  $t_b$ ,

$$dV = \frac{V \cdot \frac{1}{2^S}}{\log t_b} \quad (27)$$

Table 2-2 gives, for four different transmittances, both the predicted as well as the actual RVR errors for different scale factors in the calculation of RVR from Allard's law. The related RVR parameters are  $E_t = 2$  mile-candles,  $I = 10,000$  candles, and  $b = 60$  feet. The numerical computations for actual RVR errors were carried out for a 16-bit-wordlength integer arithmetic. The actual and predicted error

TABLE 2-2. PREDICTED AND ACTUAL ERROR IN RVR, DUE TO SCALING IN COMPUTATION OF RVR FROM ALLARD'S LAW, USING FIXED-POINT CALCULATIONS, FOR FOUR DIFFERENT TRANSMITTANCES ( $E_t = 2$  MILE-CANDLES,  $I = 10,000$  CANDLES,  $b = 60$  FEET)

Scale Factor	$t_b = 0.0108$ ; $V = 200$ ft.		$t_b = 0.341$ ; $V = 700$ ft.		$t_b = 0.53$ ; $V = 1100$ ft.		$t_b = 0.91$ ; $V = 5400$ ft.	
	$\Delta V_{pred}$ ft	$\Delta V_{act}$ ft	$\Delta V_{pred}$ ft	$\Delta V_{act}$ ft	$\Delta V_{pred}$ ft	$\Delta V_{act}$ ft	$\Delta V_{pred}$ ft	$\Delta V_{act}$ ft
32	1	2	17	52	44	162	1340	1389
64	0	1	8	23	22	35	670	1312
128	0	0	4	8	11	7	335	234
256	0	0	2	2	5	8	169	196
1024	0	0	1	3	2	8	83	13

values seem to be roughly of the same order (bearing in mind that Eq. (26) gives only the order of errors expected).

The loss of significance and, consequently, the error in RVR determination can be held down by using a scale factor as large as is feasible.

### 2.5.2 Overflow Problems

The upper limit on the magnitude of the scale factor is governed by overflow problems associated with large numbers for a limited-wordlength arithmetic.

Table 2-3 shows the magnitude of overflow problems associated with the computation of RVR. It shows the data inputs and various quantities that enter into the computation during a Newton-Raphson algorithm for Allard's law. Also shown are the associated lower and upper limits within which the values of these quantities would lie, assuming a scale factor of 1 for the quantities involving logarithms. The quantities  $V \log t_b/2b$  and  $f(V)$  may attain a value as high as  $2^s \cdot 600$ , assuming that scaling is employed for calculations. Such high values may be obtained during an iteration at low values of  $t_b$  coupled with high values of  $V$  (see Figure 2-1). High values of  $V$  may result, due to a poor initial guess for  $V$ , or may be obtained during iterations if a higher-order algorithm, such as modified Newton-Raphson or Halley's algorithm, is used with a low value assigned to  $\gamma$ .

To prevent overflow, we require  $2^s 600 < 2^{N-1}$ , where  $N$  is equal to the number of bits in a wordlength used for a signed binary representation of a number. This requires  $s < N - 10$ . Thus, for a 16-bit wordlength,  $s$  should be less than 6. It may be seen from Table 2-2 that the accuracy in RVR associated with  $s = 32$  is poor at high values of  $t_b$ . For example, at  $t_b = 0.91$  and  $V = 5400$  feet, the error in RVR is about 24%, which is a high value indeed. However, accuracy may be improved by using larger scale factors provided proper precautions are taken such that higher values of  $V$  are avoided during intermediate iterations at low values of  $t_b$ . Otherwise, it may become necessary to use larger-wordlength arithmetic or floating-point arithmetic.

A few observations on the question of overflow and convergence are considered below, before going into the consideration of measures to avoid the overflow problems.



TABLE 2-3. RANGE OF QUANTITIES INVOLVED IN CALCULATING RVR FROM ALLARD'S LAW USING NEWTON-RAPHSON METHOD

Quantity	Range	Quantity	Range	Quantity	Range
Runway visual range, V, ft	1 to 6000	log V	0 to 3.8	$f_k = 1/2 \log \frac{E_t}{I}$ - log 5280	-5.72 to -3.37
Illumination threshold, $E_t$ , mile-candles	2 to 2000	log $E_t$	0.3 to 3.3	$f_k + \log V$	-5.72 to 0.43
Intensity runway light, I, candles	400 to 20000	log I	2.6 to 4.3	$f(V) = f_k + \log V$ - $\frac{V}{2b} \log t_b$	-5.72 to 600
Atmospheric transmittance, $t_b$	$10^{-5}$ to 1	log $t_b$	-5 to 0	$\frac{V}{1 - \frac{V}{2b} \log t_b}$	1 to 10
Baseline, b, ft	50 to 750	$-\frac{V}{2b} \log t_b$	0 to 600	$\Delta V_i = \frac{Vf(V)}{1 - \frac{V}{2b} \log t_b}$	-5.72 to 6000

(1) At low  $t_b$  and high RVR

$$f(V) \approx V \log t_b/2b = \alpha,$$

and

$$f'(V) \approx \log t_b/2b = \alpha/v,$$

thus, for a Newton-Raphson method

$$\Delta V_i \approx -V_i, \text{ and } V_{i+1} \approx 0.$$

(2) Convergence to a correct value of RVR is assured using an iteration scheme of the type:

$$V_{i+1} = V_i - \epsilon_i \text{ sign}(f(V_i)), \epsilon_i > 0.$$

(3) To prevent overflow, we require:

$$2^s f(V) < 2^{N-1}, s \leq N - 5.$$

This in turn implies:

$$-2^s V \log t_b/2b < 2^{N-1} - 2^s (f_k + \log V).$$

From Table 2-3:

$$2^s (f_k + \log V) < 2^s \times 0.43 < 2^{s-1}.$$

Thus, to prevent overflow, at all stages of iteration we require

$$-2^s V \log t_b$$

to be less than  $K$ , where

$$K = 2^{N-1} - 2^{s-1}.$$

### 2.5.3 Modification to RVR Algorithm

The above discussion suggests the following modifications to the RVR algorithm for Allard's law described in Subsection 2.3:

(1) Set  $-V \cdot 2^s \log t_b/2b = K$ , if  $-V \cdot 2^s \log t_b/2b > K$ , with

$$K = 2^{N-1} - 2^{s-1}$$

and

$$s \leq N - 5.$$

(2) Set  $\beta = \gamma$  if  $\beta < \gamma$ ,  $\gamma > 0$

for modified Newton-Raphson and Halley's algorithms.  
The value for  $\gamma$  is to be appropriately selected to assure convergence without any overflow problems that may arise at a low value of  $\gamma$ .

It is easy to ascertain, in the light of earlier observations, that the convergence of the modified RVR algorithm for Allard's law is assured without any overflow problems.

#### 2.5.4 Wordlength and RVR Accuracy

With the modifications incorporated into an RVR algorithm as discussed in Subsection 2.5.3, higher accuracies in RVR can be achieved for a given wordlength by using larger scale factors.

Figures 2-2 to 2-4 show the effect of limited wordlength on the accuracy of RVR calculated from Allard's law using fixed-point arithmetic and a scale factor of  $2^s$  with  $s = N - 6$ . Figures 2-2 and 2-3 correspond to nighttime conditions ( $E_t = 2$  mile-candles) whereas Figure 2-4 corresponds to bright daytime operations ( $E_t = 2600$  mile-candles). The runway light intensity,  $I$ , is equal to 10,000 candles and the transmissometer baselength is equal to 60 feet.

Figures 2-3 and 2-4 show the percentage errors in transmissivity that arise due to truncation in the calculation of its logarithm employing 12-, 14-, and 16-bit-wordlength arithmetic.

Besides the errors due to truncation during RVR calculations, additional errors arise in transmittance during its measurement and transmission. The accuracy required in the calculation of RVR is further governed by the decision, control, and display requirements. Thus, the question of the choice of a suitable wordlength for the signal data converter is integrally related to the rest of the RVR system as well. However, it is possible to make several observations on the wordlength required, without having to consider the accuracy of the rest of the RVR system. Thus, 12-bit wordlength for RVR calculations may be ruled out due to high errors and inadequate range ( $-2048 \leq \text{number} \leq 2047$ ) available for calculations. On the other hand, 16-bit wordlength seems to be cost-effective in providing adequate accuracy for RVR calculations and adequate range ( $-32768 \leq \text{number} \leq 32767$ ) for the representation of a number. A recent survey of minicomputers<sup>(1)</sup> indicates that more than 80% of the minicomputers available in the market have a 16-bit wordlength.

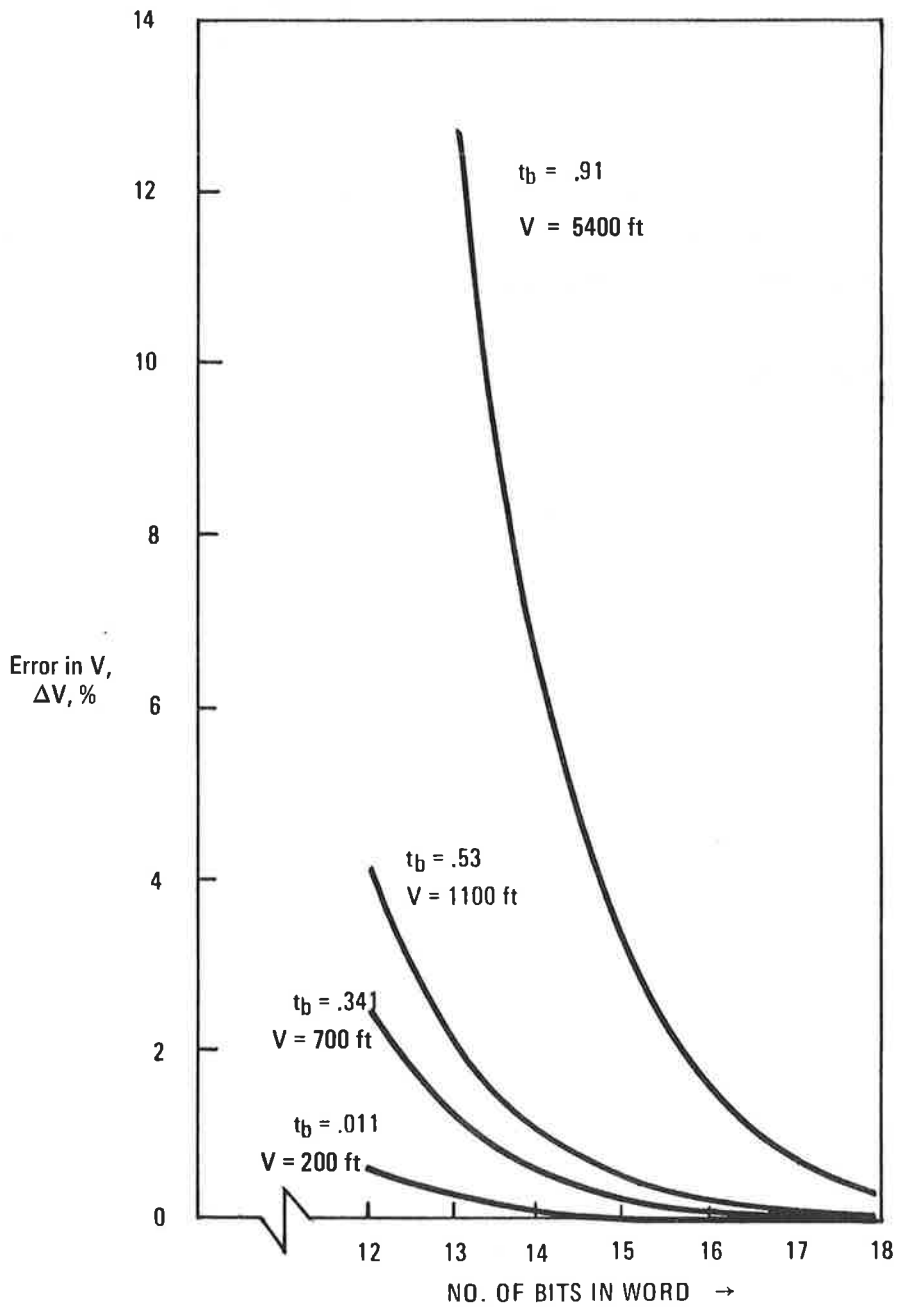


Figure 2-2. Truncation error in RVR due to limited wordlength in fixed-point arithmetic calculations of RVR using Allard's law for four different values of atmospheric transmittances  $t_b$  (and associated correct RVR) with  $E_t = 2$  mile-candles,  $I = 10,000$  candles, and baselength  $b = 60$  feet.

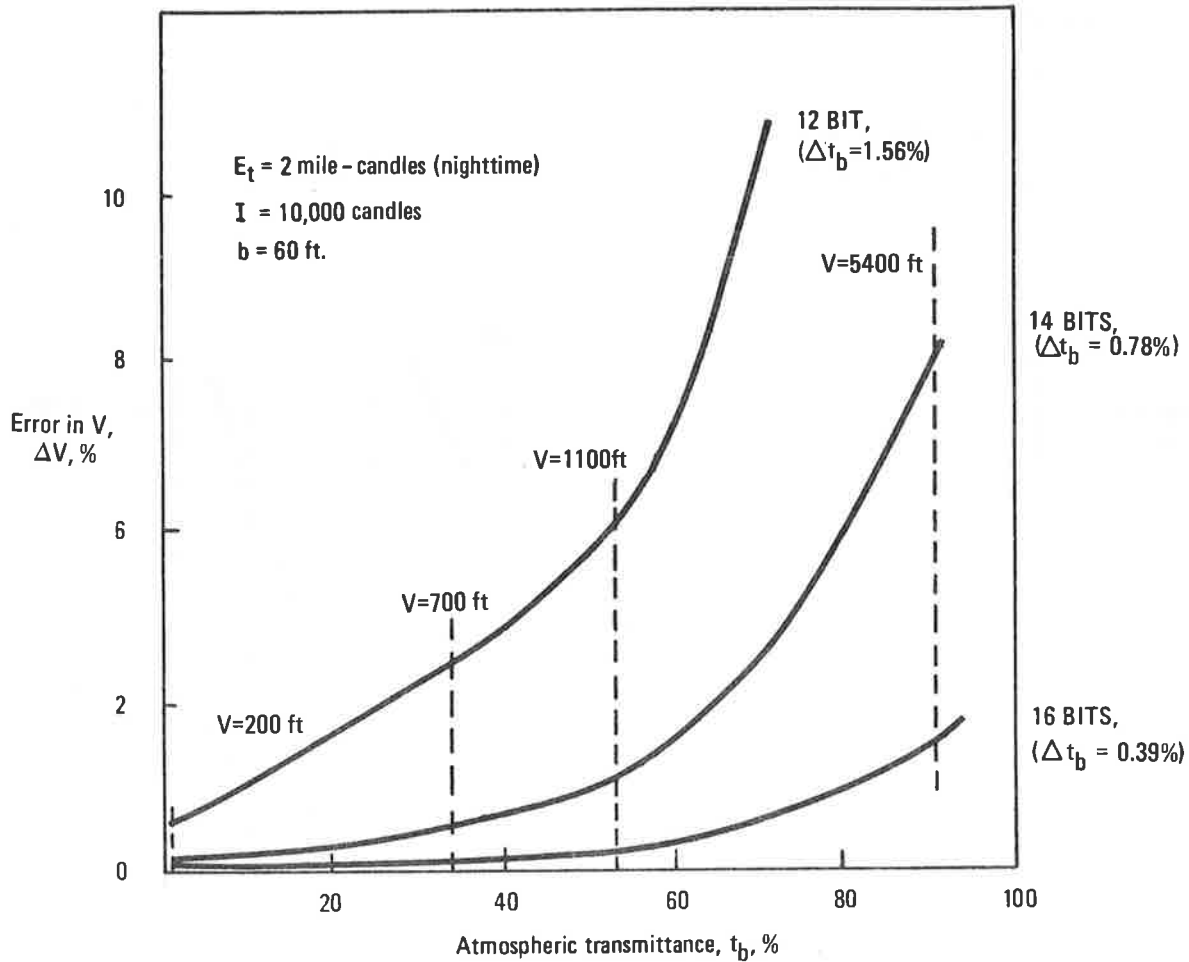


Figure 2-3. Plot of  $\Delta V(\%)$  vs  $t_b(\%)$  for nighttime conditions showing the dependence of truncation error in the calculation of  $V$  on the atmospheric transmittance  $t_b$  for four different word-bit lengths.

## 2.6 SUMMARY AND CONCLUSIONS

Different aspects related to the determination of RVR have been discussed in this section. Efficient algorithms are developed for the computation of RVR from Allard's and Koschmieder's laws using fixed-point limited-wordlength calculations. It was shown that the order of accuracy attainable for a given wordlength is limited by two factors: errors due to truncation in the computation of the logarithm of atmospheric transmittance, and the overflow problems associated with limited wordlengths and large numbers. It was shown that the accuracy in the determination of the logarithm of a number using fixed-point arithmetic is of the order of  $2^{N-5}$ , where  $N$  = number of bits in the signed binary

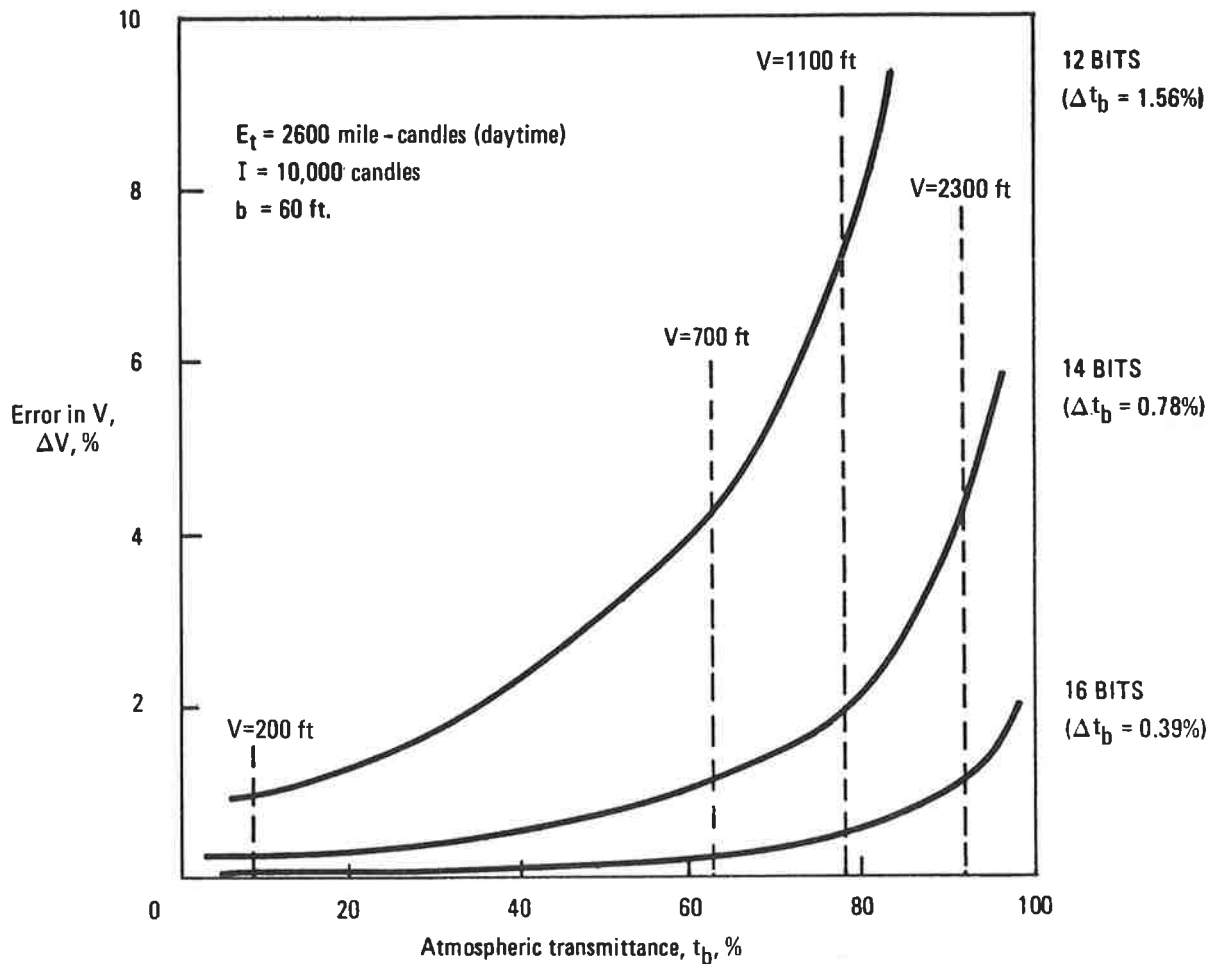


Figure 2-4. Plot of  $\Delta V(\%)$  vs  $t_b(\%)$  for daytime conditions showing the dependence of truncation error in the calculation of  $V$  on the atmospheric transmittance  $t_b$  for four different word-bit lengths.

representation of a number. Relations were derived to project the order of accuracy available for a given visual environment and transmissometer baselength. It was seen that 12-bit wordlength does not meet the range requirements for the representation of a number and, moreover, the accuracy attainable with such a wordlength may be unacceptable. A 16-bit wordlength provides adequate range for the representation of a number and the accuracy available in the determination of RVR seems acceptable. Suitable measures necessary to achieve the projected order of accuracy, global convergence, and prevention of overflow during calculations were discussed for incorporation in the RVR algorithms.

### 3. VISIBILITY SYSTEM SOFTWARE

#### 3.1 INTRODUCTION

This section describes the basic software modules required to implement a new airport visibility measuring and data-acquisition system described in more detail in Reference (1).

The design of the system software accounts for a variety of system configurations that may exist at different airport installations now and in the future. It is envisaged that the software package for a particular installation could be assembled by putting together the relevant software modules for each component and phase of operation of the system.

#### 3.2 BASIC SOFTWARE PACKAGE

The basic software package is shown in Figure 3-1. It consists of three primary modules—VISIB, LKSERV, and EXECUT and two supporting modules ARITH and IOPKG. VISIB is a basic module that handles input, output, control, and calculations for the visibility system installation with the support of LKSERV, EXECUT, ARITH, and IOPKG.

LKSERV is primarily a time-keeping routine, servicing the interrupt requests generated by the real-time clock. It automatically updates the time of day and date after they are initially entered into the software from the executive program EXECUT. It is also used to clock I/O operations such as the transmissometer pulse count. LKSERV also provides a software interrupt to flag a status change in the runway-selector switch located on the display console.

EXECUT allows real-time software flow control through teletype command and response repertoire. Included is a test mode to check the modules VISIB, ARITH, and IOPKG. Visibility parameters are entered through the keyboard for test purposes. A diagnostic printout on the teletype is included. EXECUT also provides the necessary flexibility

to update basic visibility parameters through a teletype input mode. Commands to interrogate the current time and date are included.

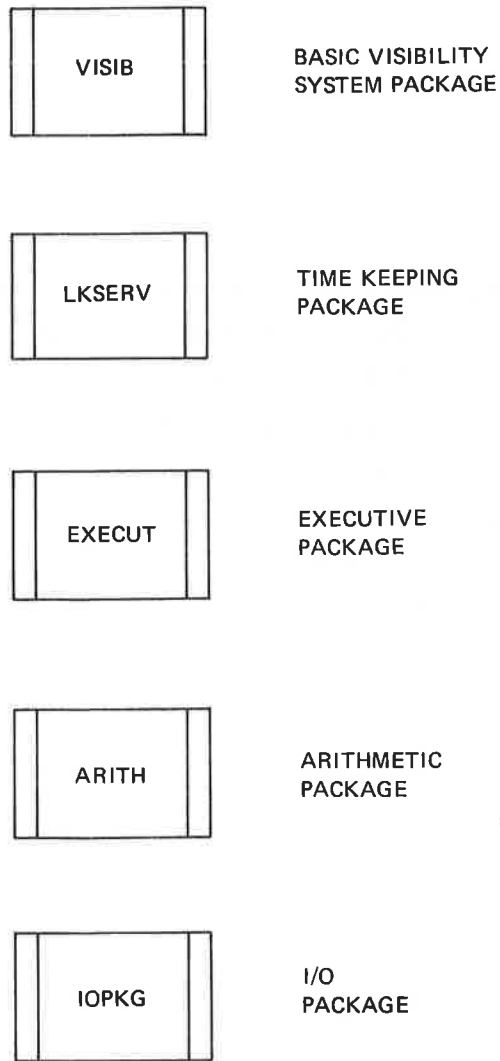


Figure 3-1. Primary modules comprising an airport visibility system software developed by the Draper Laboratory.

ARITH is a software math package to implement those functions not handled by hardware. Included are fixed-point multiply, divide, logarithm, and normalize. The ARITH module described in Appendix C was specifically developed for the experimental simulator-interface described in Section 5.



IOPKG is a software module to handle interface I/O communication. Appendix D contains a description of TTYPKG developed to permit I/O in various formats for a teletype interface.

These software modules are described in detail in the following subsections.

### 3.3 VISIB

VISIB constitutes the backbone of the airport visibility system software. As shown in Figure 3-2, it consists of four modules serving the following functions:

- (1) Data input from visibility system sensors.
- (2) Calculations to evaluate runway visual range (RVR), slant visual range (SVR), taxiway visual range (TVR), and ceiling in advanced modules.
- (3) Output to displays, data logging, and system controls.

These four modules service more than one external device or sensor. For example, INPUT handles input data from the transmissometers, the TVR, SVR, and ceilometer sensors, and runway light intensity setting, etc. The modules consist of submodules, each handling a specific device or sensor. Dummy submodules replace the original when the device or sensor is nonexistent. New or replacement sensors are easily provided for through submodular software addition or replacement.

Subsections 3.3.1 to 3.3.4 further describe the four modules constituting VISIB. The software submodules associated with the evaluation of SVR, TVR, and ceiling, are sketched only briefly, since these systems and associated calculations are currently in various stages of investigation and development and consequently not well defined. In all probability, the data reduction for SVR and possibly ceiling will be performed by dedicated instrumentation and the result simply handed on to the above-referenced software for subsequent display. By contrast, the operations required to reduce transmissometer data to visibility information is well documented and based upon the software described in Appendix C for the experimental simulator-interface.

#### 3.3.1 INPUT

INPUT (see Figure 3-3) reads data from external sensors and devices and is stored until new data is read during the next input cycle. The operations of the submodules are explained in the following: TRIN (see

Figure 3-4) handles transmissometer pulse-rate data input. The atmospheric transmittance is averaged over a period of TW seconds. This is done by a pulse count over this period, handled by interface hardware. Alternative transmission modes of transmittance data would require a software modification.

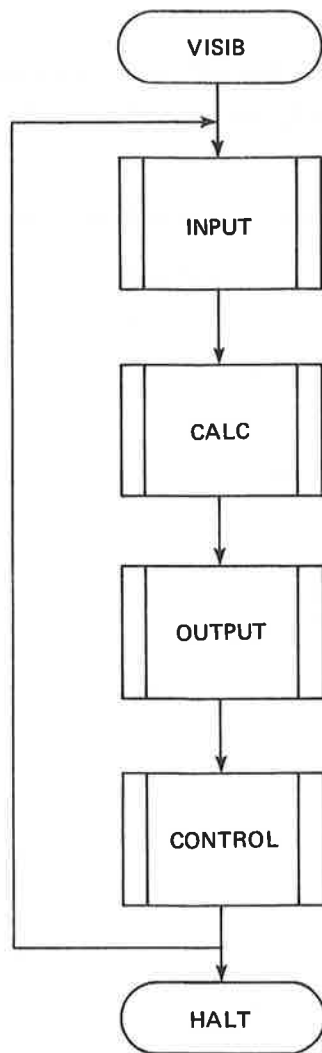


Figure 3-2. Basic airport visibility system package which handles during an update cycle: data inputs, calculations, transmission, recording display, and control functions associated with the system.

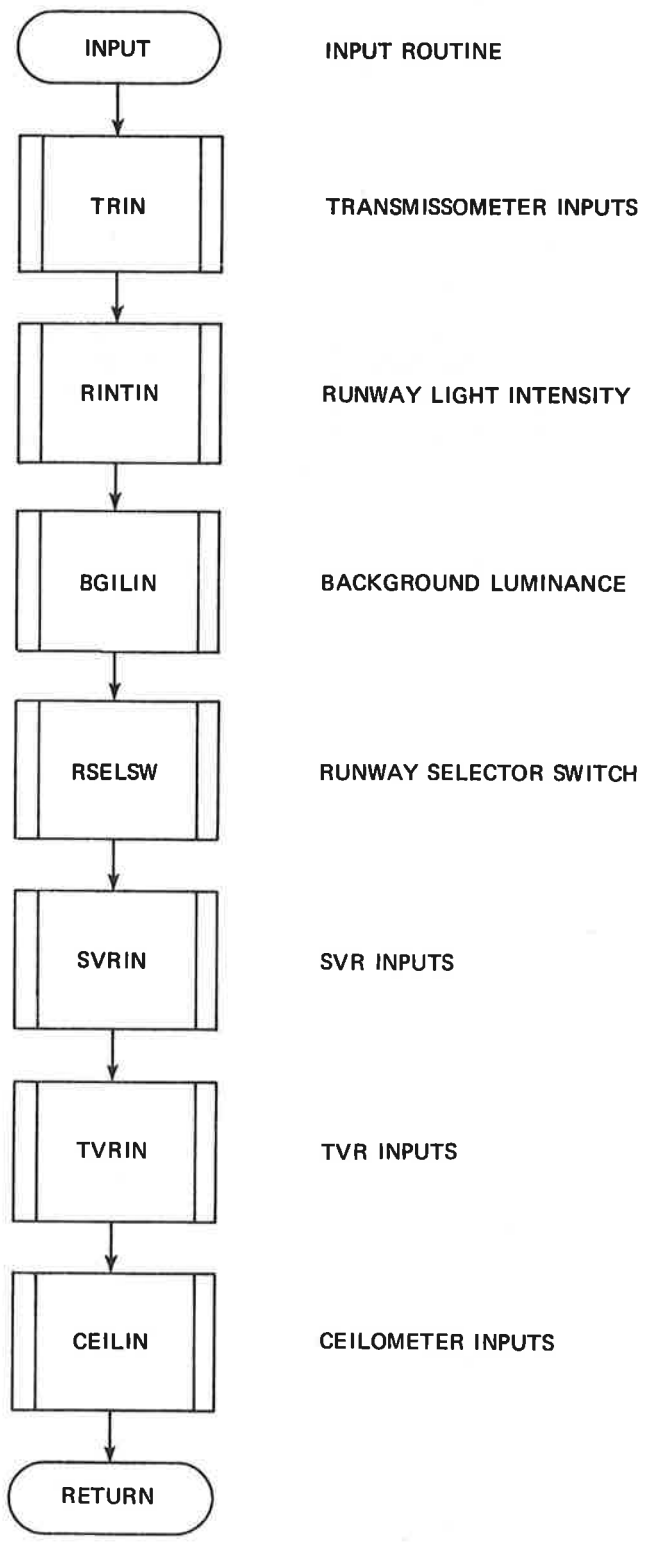


Figure 3-3. Routine to input data.

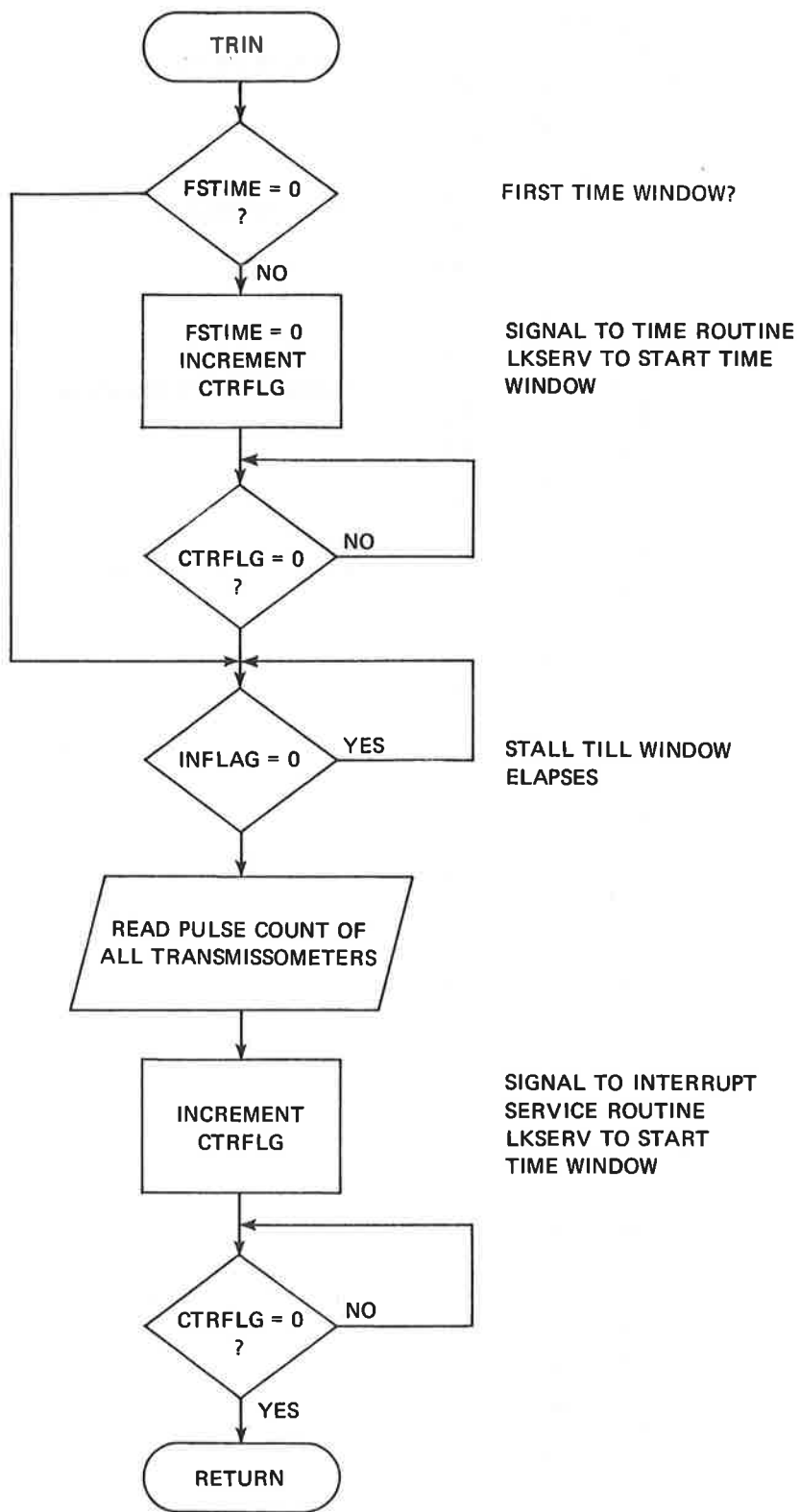


Figure 3-4. Transmissometer data input routine.

TRIN also decides when to start the time window for the pulse count. The actual enabling and inhibiting of the pulse count is handled by the time-keeping routine LKSERV, which services the real-time clock interrupt. Communication between TRIN and LKSERV is achieved through the flags CTRFLG and INFLAG. Setting CTRFLG flags LKSERV to enable the pulse count. LKSERV inhibits the count after TW seconds, and sets the flag INFLAG. When the pulse count begins, the VISIB program flow resumes through the basic cycle, INPUT, CALC, OUTPUT, and CONTROL, before returning to INPUT and TRIN for the start on the next time window. If the previous time window has elapsed (determined by the status of INFLAG), the pulse count of all transmissometers is read and the flag CTRFLG set to flag LKSERV to enable the pulse count once again. If the time window has not elapsed, the program waits. A provision has been made to prolong this waiting period every time VISIB is entered (such as at start-up or from EXECUT) when no previous pulse count is available. The timing relationship of these operations is shown in Figure 3-5.

RINTIN (see Figure 3-6) establishes the runway-light intensity for each runway, reading data from the intensity setting switch or from a light-intensity monitor. The nominal light intensities associated with each setting are available in the software.

BGILIN (see Figure 3-7) establishes the visual illuminance threshold (in mile-candles) either as a discrete level or a continuous measurement.

RSELSW (see Figure 3-8) reads the runway-selector switch on the display console. SVRIN, TVRIN, and CEILIN (see Figures 3-9 through 3-11) read data from the slant visual range, taxiway visual range, and ceilometer instrumentation.

### 3.3.2 CALC

CALC (see Figure 3-12) processes the visibility system sensor data to provide a measure of RVR, SVR, TVR, and ceiling, characterizing the visual environment.

RVRCAL (see Figure 3-13) evaluates the RVR corresponding to each measurement of transmittance, background luminance, and runway-light intensity. The RVR values are stored until recalculated in the next cycle. RVRCAL is serviced by ITER (see Figure 3-14) which evaluates RVR using either Allard's Law or Koschmieder's Law.

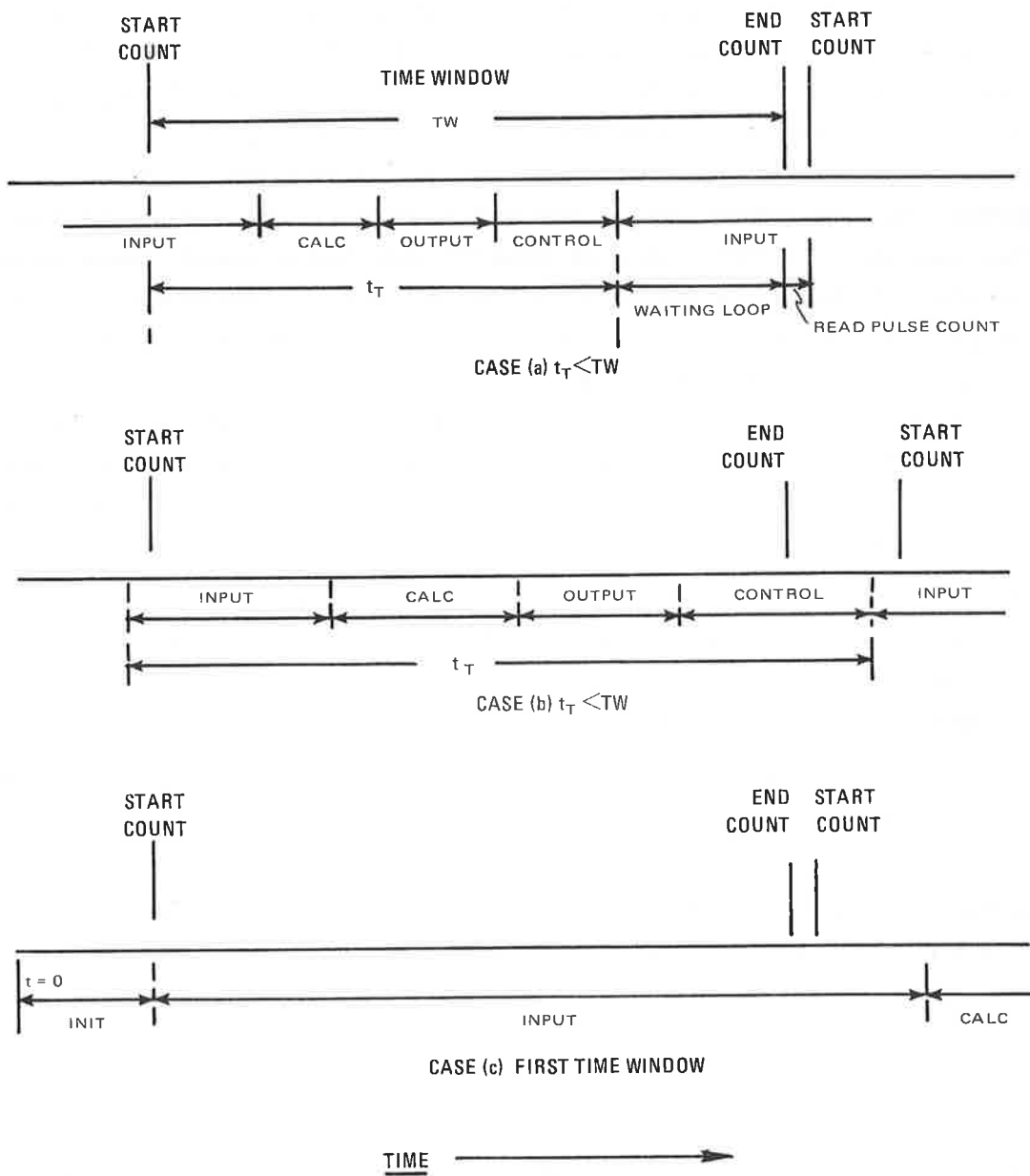


Figure 3-5. Timing relationships for different software modules with respect to the transmissometer pulse counting during an update cycle.

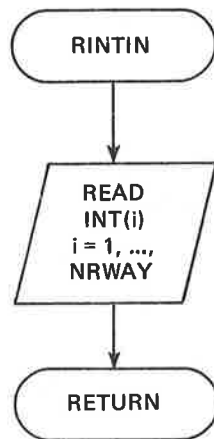


Figure 3-6. Runway-light-intensity input routine.

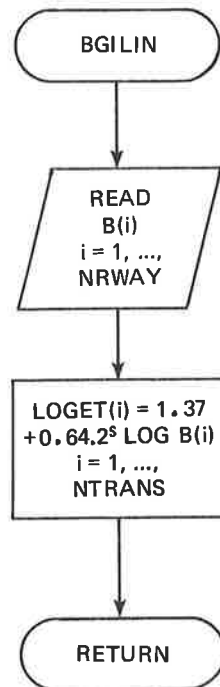


Figure 3-7. Sky-background-luminance input routine.

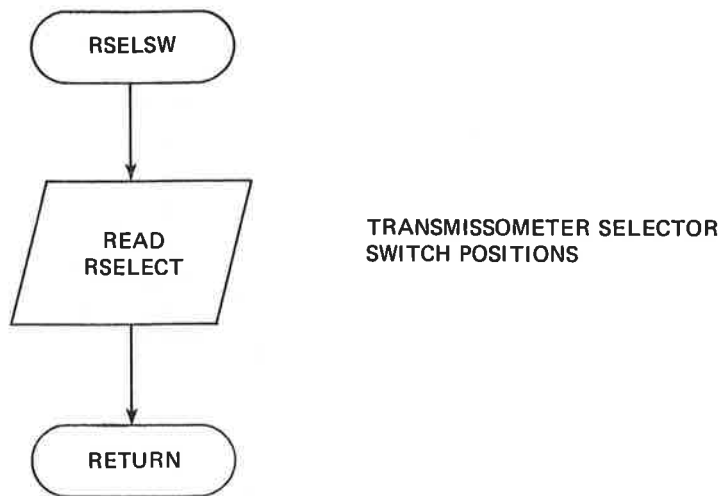


Figure 3-8. Routine to read transmissometer selector switch position for RVR display.

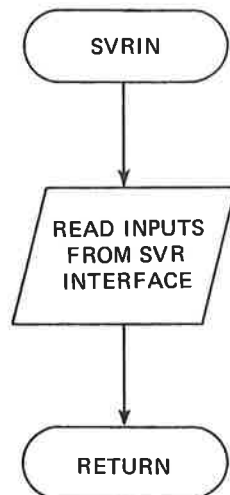


Figure 3-9. Routine to read inputs from SVR interface.



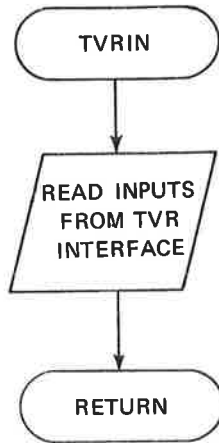


Figure 3-10. Routine to read TVR inputs.

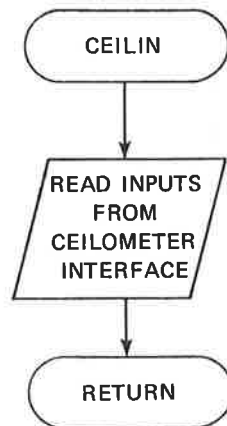


Figure 3-11. Routine to read ceilometer inputs.

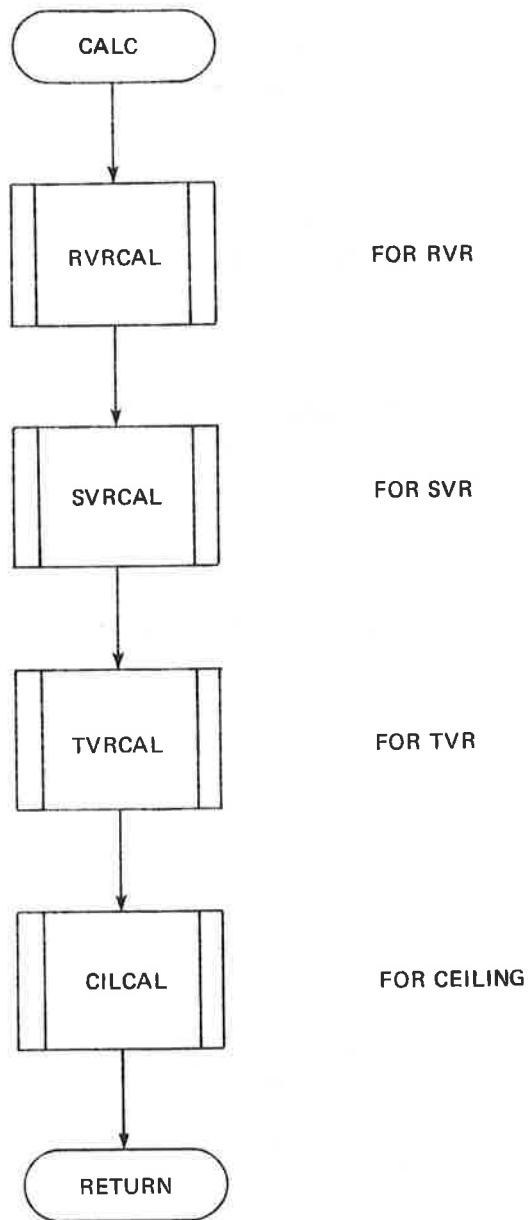


Figure 3-12. Airport visibility system parameters calculation routine.

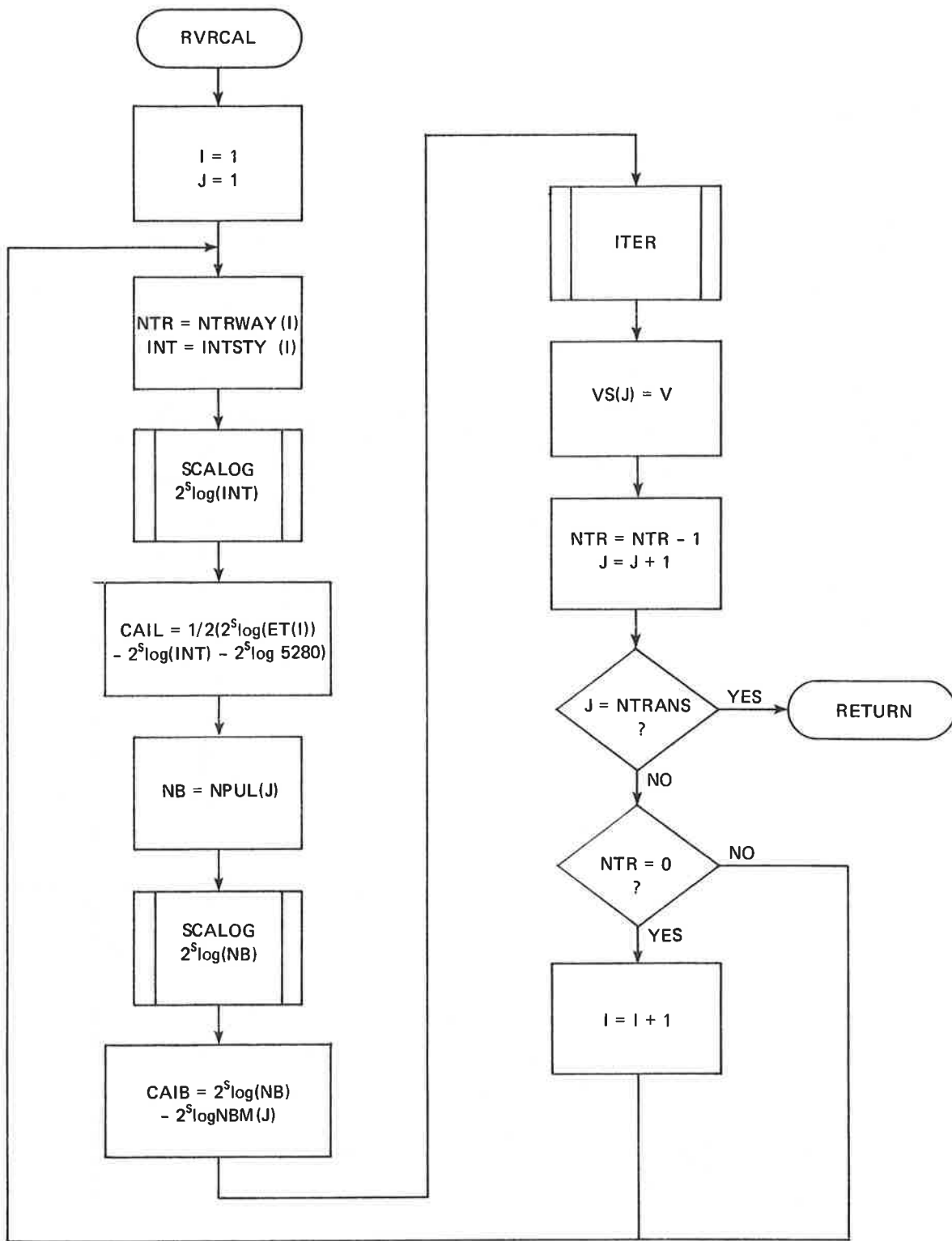


Figure 3-13. Routine to compute RVR.

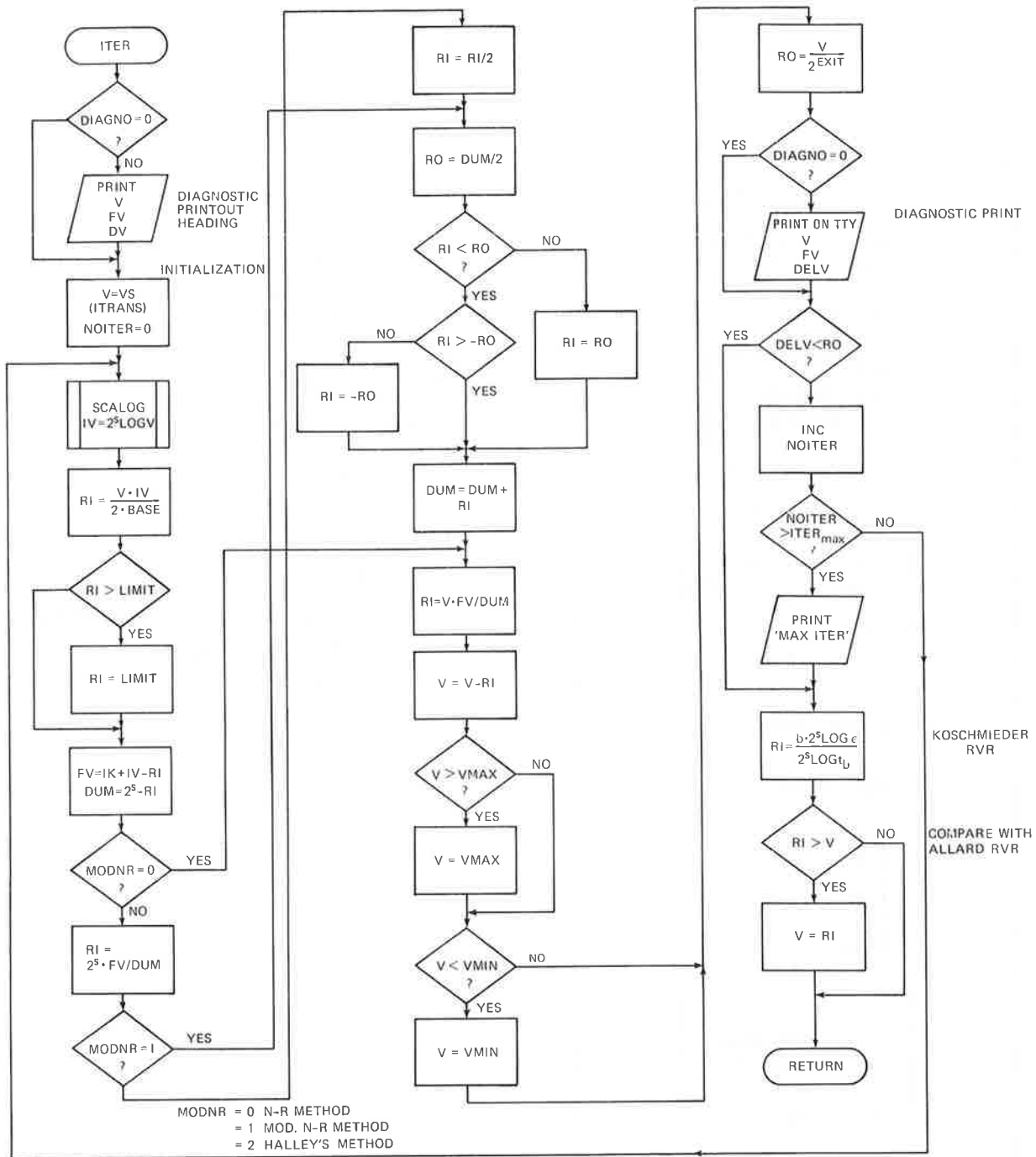


Figure 3-14. Iteration routine to compute RVR.

ITER evaluates RVR in fixed-point arithmetic with a scale factor of  $2^{**SCALE}$ .<sup>\*</sup> An algorithm based on Allard's Law requires an iterative solution; the maximum number of iterations is ITERMAX. If the solution has not converged before the limiting number of iterations ( $\Delta V_i > V_i / 2^{**EXIT}$ ), an error message is generated. Three iterative methods are coded, namely Newton-Raphson (N-R), modified N-R, and Halley's method. The software switch MODNR selects the desired algorithm. The iterative algorithms are described in Subsection 2.3. The algorithms have been carefully structured to ensure global convergence and maximum accuracy for the wordlength.

SVRCAL, TVRCAL, and CILCAL (see Figures 3-15 through 3-17) are the submodules supplying SVR, TVR, and ceiling.

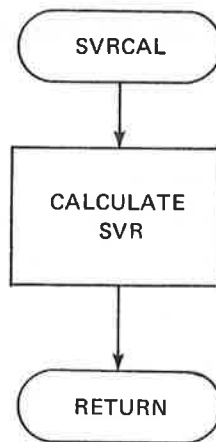


Figure 3-15. Routine to compute SVR.

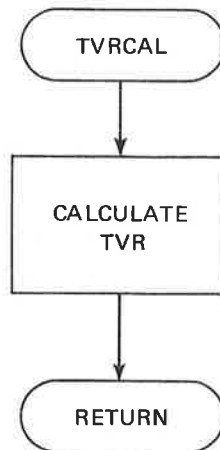


Figure 3-16. Routine to compute TVR.

---

\* Symbol \*\* stands for exponentiation.

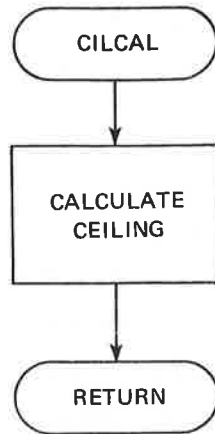


Figure 3-17. Routine to compute ceiling.

### 3.3.3 OUTPUT

OUTPUT (see Figure 3-18) transmits the processed visibility information to local and remote displays, data logging system, and data up-link transmitter. OUTPUT (see Figure 3-19) submodules are as follows.

OUTATC transmits visibility data to display consoles. The specific data transmitted depends upon the position of the runway-selector switch. The software interrupt routine LKSERV interrogates the position of the runway selector every second. If a change has occurred, the new visibility data is immediately transmitted to the displays by STROBE (see Figure 3-20).

ATCRVR, ATCSVR, ATCRVR, and OUTREC (see Figures 3-21 through 3-29) are the submodules displaying SVR, TVR, and ceiling information.

OUTREC assembles an output list for data logging according to a specified format, and outputs the data to the teletype or other data-logging medium.

OUTITS assembles an output list according to a specified format for output to an information-transmission system.

### 3.4 LKSERV

LKSERV (see Figure 3-30) is one of the three main packages referred to in Figure 3-2. It is basically a time-keeping routine

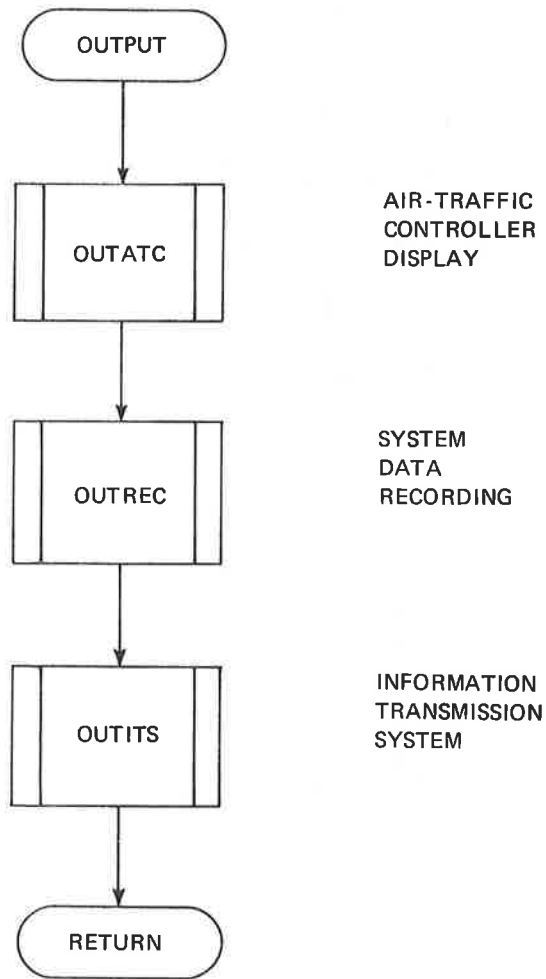


Figure 3-18. Output routine.

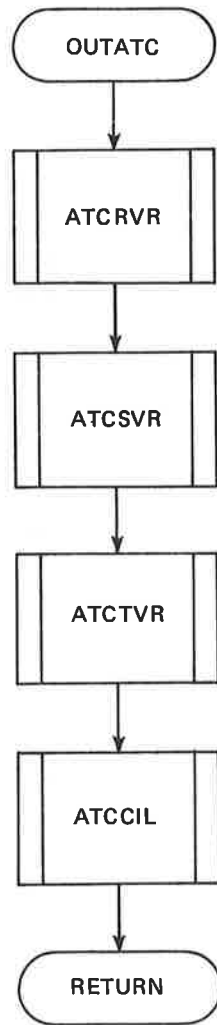


Figure 3-19. Routine to service air-traffic-controller display.



which operates at a higher priority level than the other software routines. LKSERV uses the hardware line-frequency real-time clock. When the latter generates an interrupt, control is transferred to LKSERV; upon return from LKSERV, the main program resumes from the point of interruption.

LKSERV is also used to enable and inhibit the transmissometer pulse count. The routine inhibits the count when the time window has reached the specified width, TW seconds. Initiation of the time-window occurs in TRIN (a submodule of VISIB) and is communicated to LKSERV by a flag.

LKSERV also includes a software interrupt to service a display update request signalled by the runway-selector switch. LKSERV interrogates the switch status every second. If a change has occurred in the last record, control is briefly passed to STROBE (see Figure 3-20). The latter strobes (using OUTATC) the new information to the displays, then a return is made to LKSERV. STROBE operates at a lower priority level than LKSERV.

The time and date are automatically updated while the program runs uninterrupted. A restart requires an initialization of time and date. The accuracy of the time and date is that of the line-frequency real-time clock.\*

### 3.5 EXECUT

EXECUT (see Figure 3-31) provides real-time monitor control of the software from the keyboard. The monitor is entered by hitting the control-C key on the keyboard. EXECUT acknowledges by typing the character '.' on the teletype.\*\* A string of characters terminated by

---

\* The execution time of LKSERV must be less than the interrupt period, otherwise the time-keeping ability of LKSERV would be lost. This is not a severe limitation, since the functional requirements of LKSERV which determine the execution time are minimal.

\*\* IOPKG services the interrupt generated by keyboard inputs and passes control to EXECUT when the control-C key is hit. The details may be found in TTYIO of the experimental simulation software described in Appendix D.

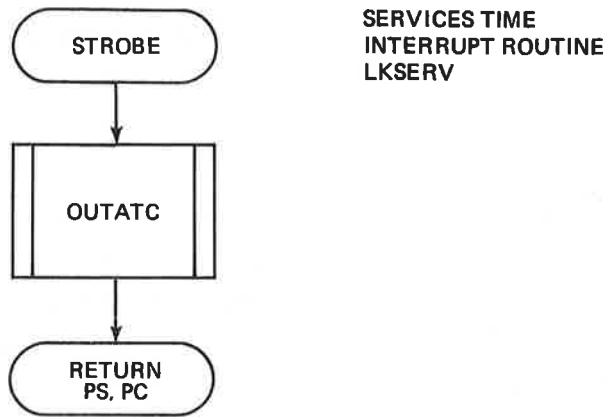


Figure 3-20. Routine to service air-traffic-control display.

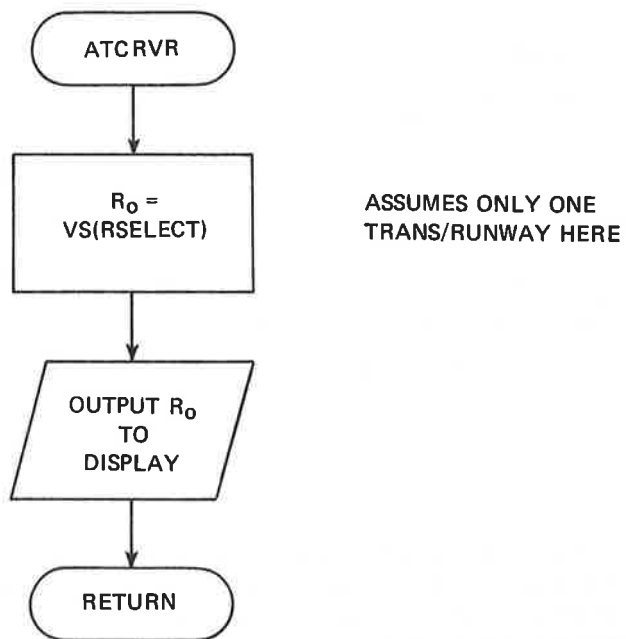


Figure 3-21. Routine to display RVR.

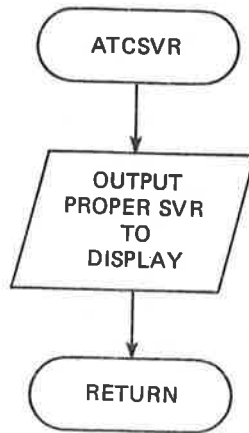


Figure 3-22. Routine to display SVR.

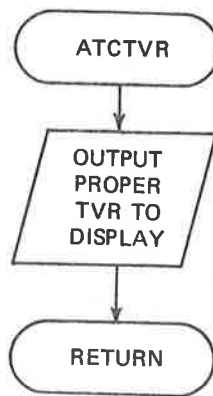


Figure 3-23. Routine to display TVR.

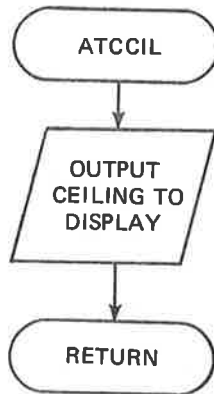


Figure 3-24. Routine to display ceiling.

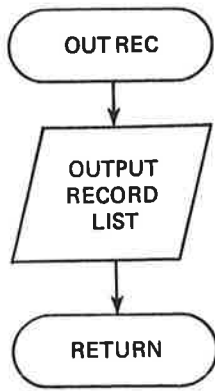


Figure 3-25. Routine for system data recording.

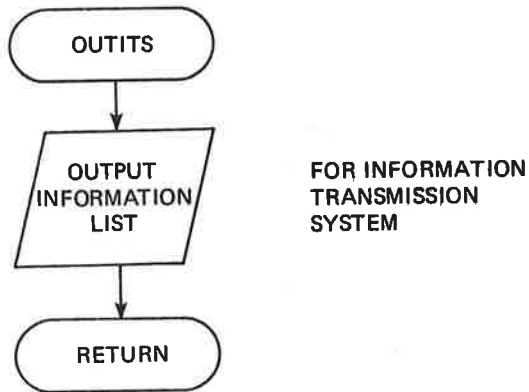


Figure 3-26. Routine for information transmission system.

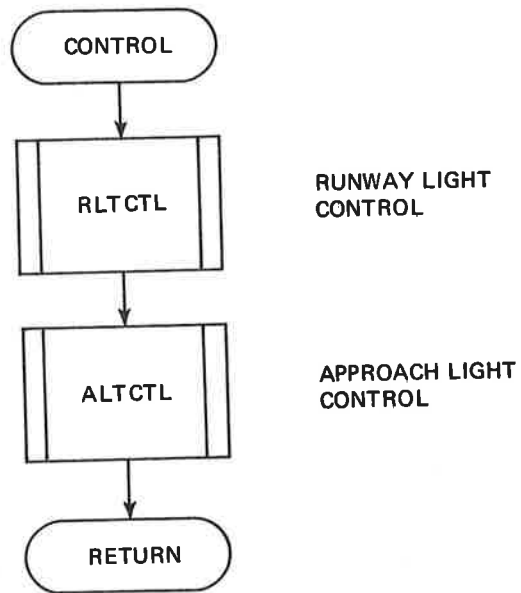


Figure 3-27. Light control routine.

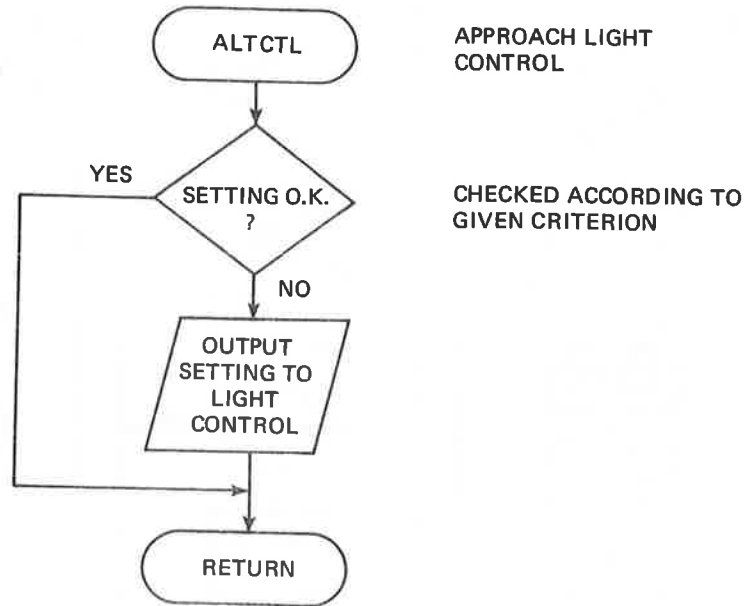


Figure 3-28. Approach light control routine.

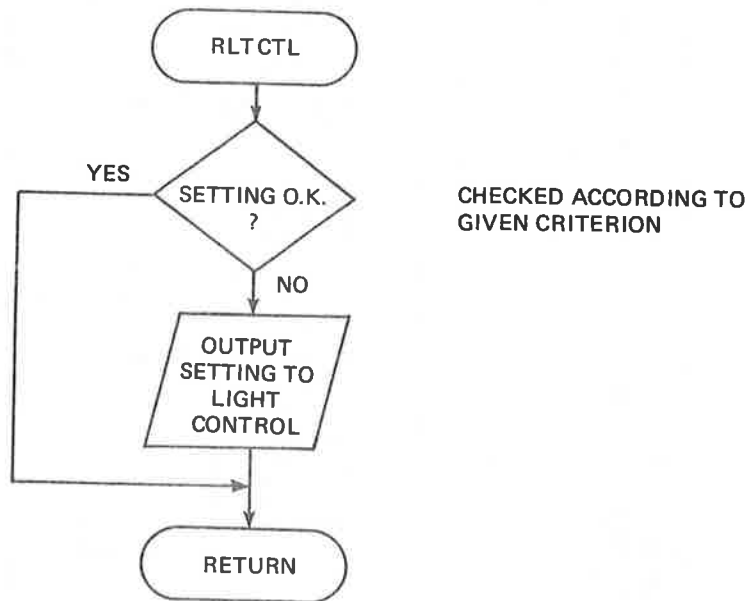


Figure 3-29. Runway light intensity control routine.

SERVICES LINE  
CLOCK INTERRUPT

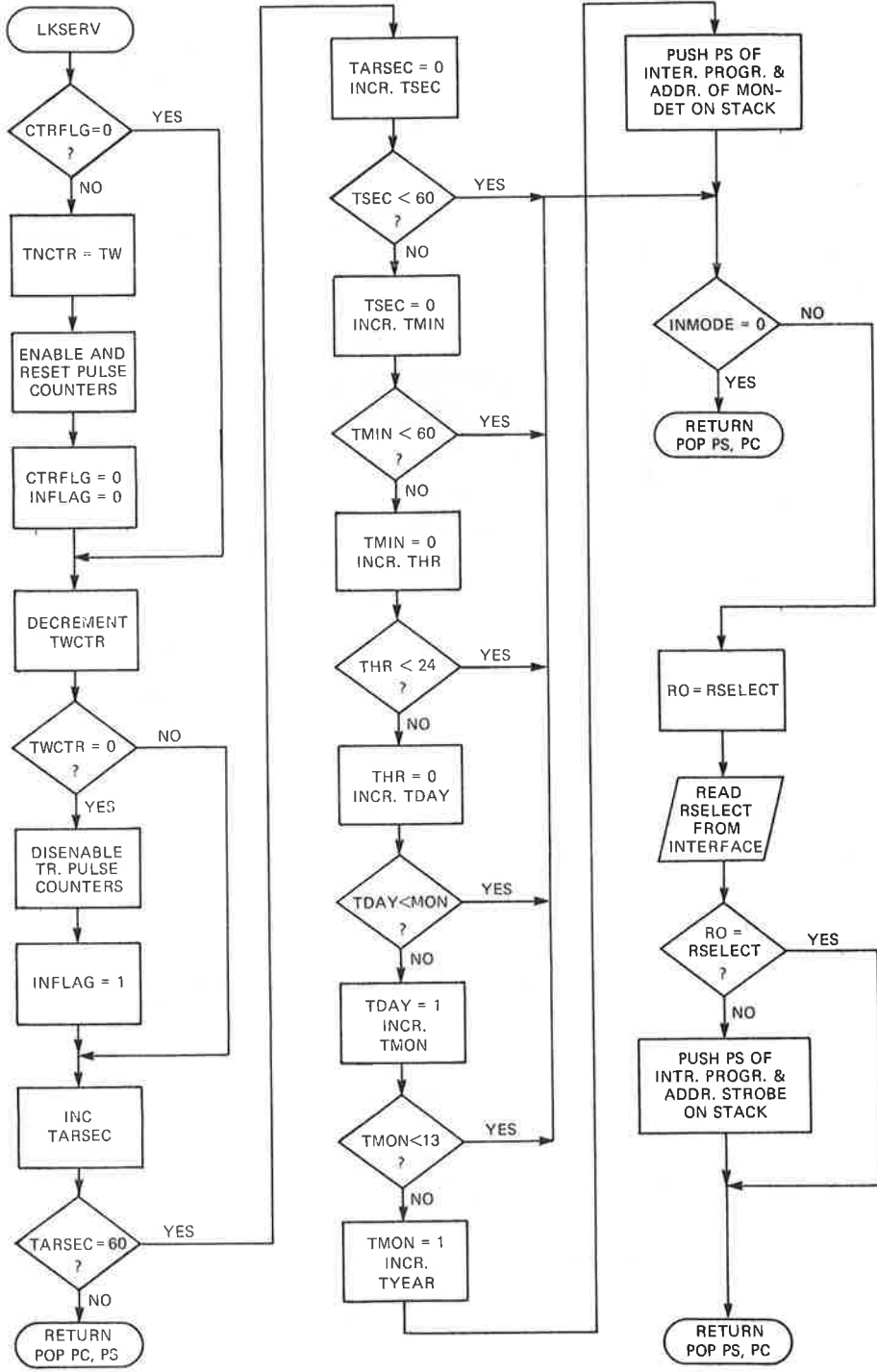


Figure 3-30. Routine to keep time and service time-related control functions.

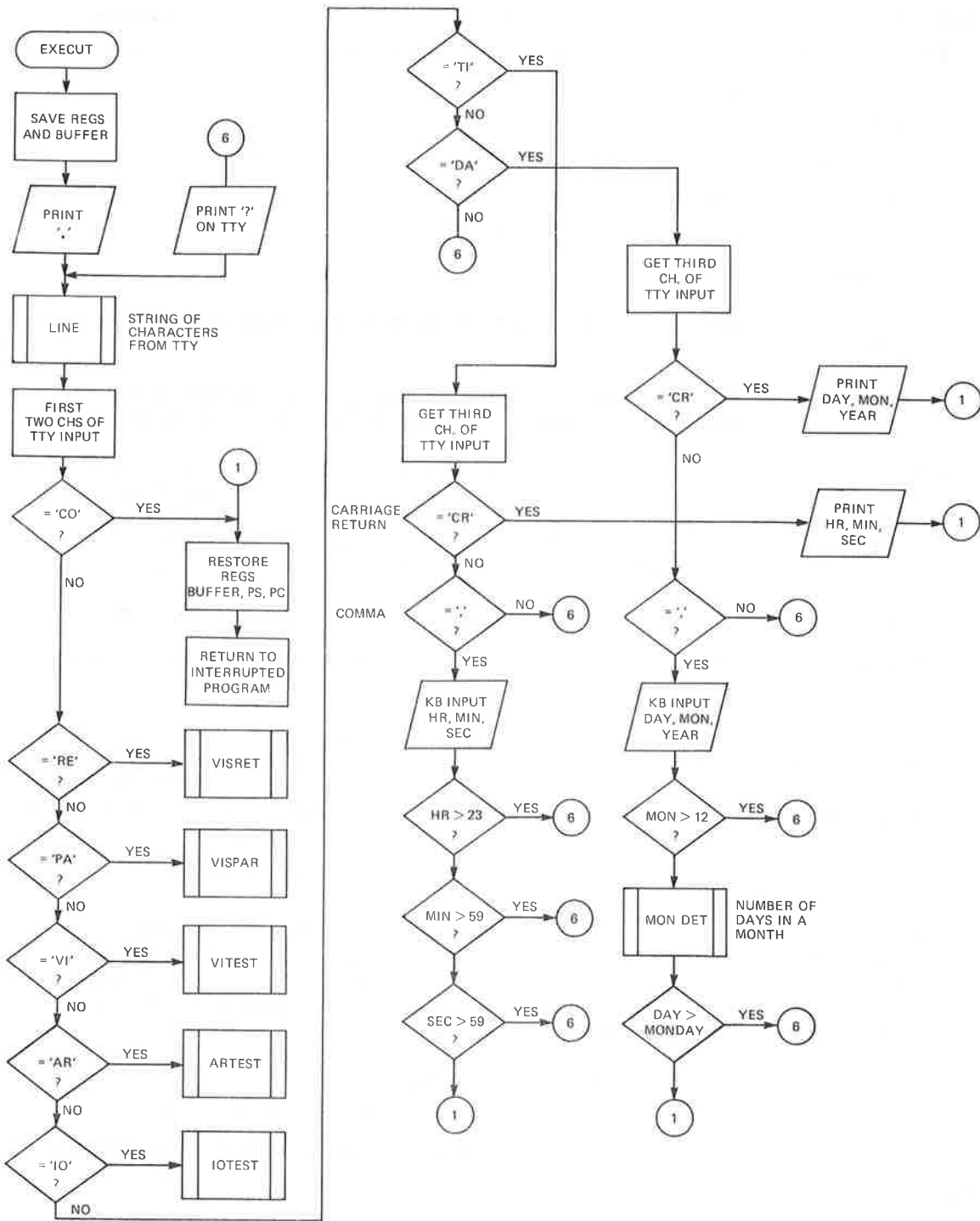


Figure 3-31. Executive routine.

a comma or carriage return (referred to in the flow diagrams as CR or ↵). can be input from the teletype. If the command is not recognized as one of those shown in Table 3-1, EXECUT prints '?' on the teletype and returns control to the interrupted program. The monitor mode must be reentered for further communication.

TABLE 3-1. EXECUT COMMANDS AND RESPONSES

Command	Response
CO	Continue with the interrupted program.
RE	Transfer to VISRET to set flags for the normal operating mode before return to VISIB.
PA	Transfer to PARASET to input parameter values from the teletype for the visibility system, and then return to VISIB to resume system operations.
VI	Transfer to VITEST, a test program to check out VISIB. Flags are set to transfer into the visibility-simulation mode, entering system data from the teletype. A diagnostic flag is also set to enable the diagnostic printout on the teletype. Control is returned to VISIB, which continues to operate in the test mode until interrupted again by the monitor.
AR	Transfer to ARTEST to test the arithmetic package from the teletype.
IO	Transfer to IOTEST to test the input/output routine IOPKG from the teletype.
TI,	Input time (hour, minute, second) from the teletype with hour <24, minute <60, and second <60. Control then returns to the interrupted program. If the constraints on the input time variables are violated, the teletype responds with '?' before returning to the interrupted program. The monitor (EXECUT) must be reentered for a new attempt to input time.
TI↵	Prints time (hour, minute, second) on teletype.
DA,	Inputs date (day, month, year) from the teletype with the day ≤MONDAY where MONDAY equals the maximum number of days for the given month and year (as computed by MONDET) and with the month ≤12. Control is then returned to the main program. If the input data violates the constraints, the teletype response is similar to that for an invalid time input command.
DA↵	Prints date (day, month, year) on the teletype.

Table 3-1 describes the EXECUT commands and program responses. The commands provide the following capabilities:

- (1) Software programs to test the modules IOPKG and VISIB.\*  
Control is returned to the interrupted program by reentering the monitor mode and issuing the return command, RE.

\* See Appendix D for details of these test programs.



- (2) Basic airport visibility system parameters may be changed by entering new values through the teletype. Control returns to the start of the main program VISIB after the new values are entered.
- (3) Time and date may be interrogated through a teletype entry. Initialization of time and date is also through the teletype. Control is returned automatically to the main program after the time and date have been typed on the teletype.

The functions of the submodules VISPAR, VISRET, VITEST, and MONDET (see Figures 3-32 through 3-35) are delineated in Table 3-1 and require no further elaboration except in the case of VISPAR. PARSET, the routine and services VISPAR, is described next.

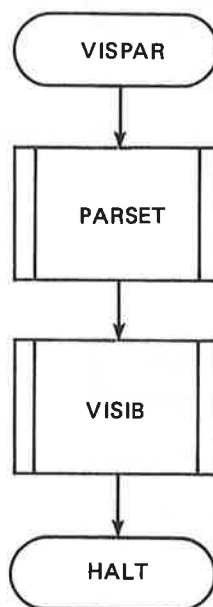


Figure 3-32. Routine to enter basic system parameters.

PARSET sets up the system and software parameters which are basic to a visibility-measuring installation, i.e., the parameters that are essentially fixed for a specific equipment configuration. These parameters are software variables that may be altered at any time during program operation. The parameters may be entered through a convenient input medium such as the teletype or through the computer console. Several submodules comprise PARSET (see Figure 3-36) and these are briefly described in the following.

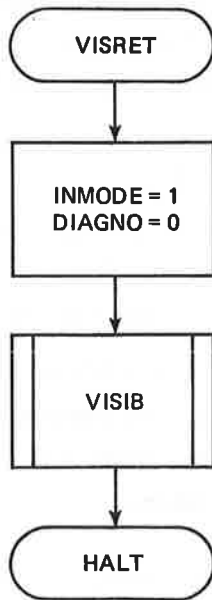


Figure 3-33. Routine for transfer to normal operating mode.

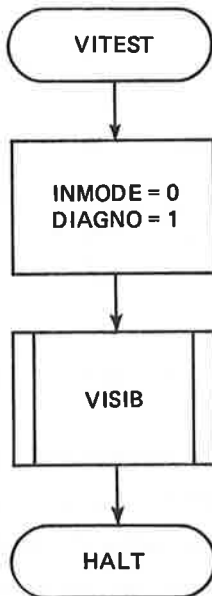


Figure 3-34. Test routine to check out VISIB.

DETERMINES MONDAY  
(NUMBER OF DAYS IN  
A MONTH + 1)

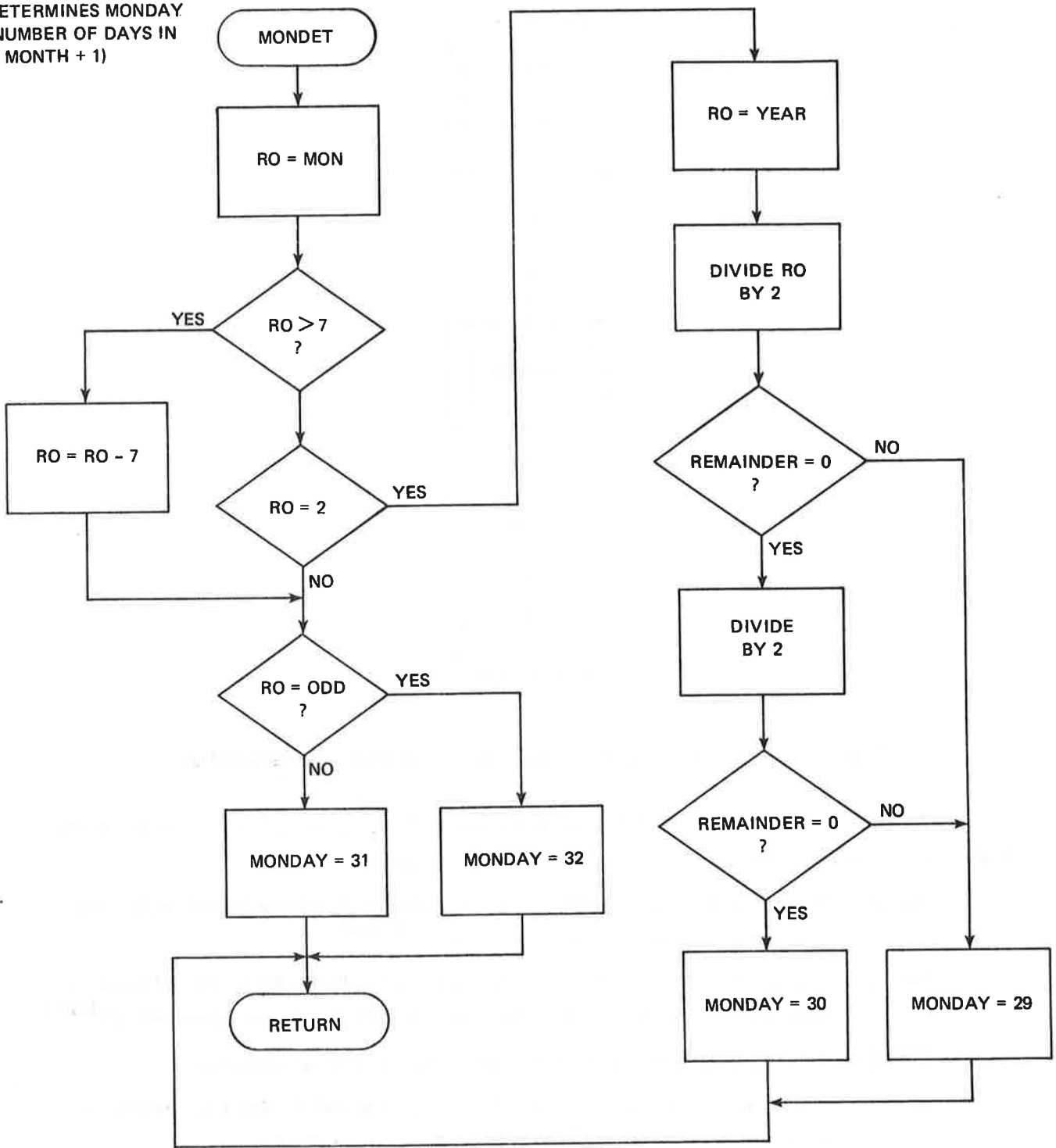


Figure 3-35. Routine to determine number of days in a month.

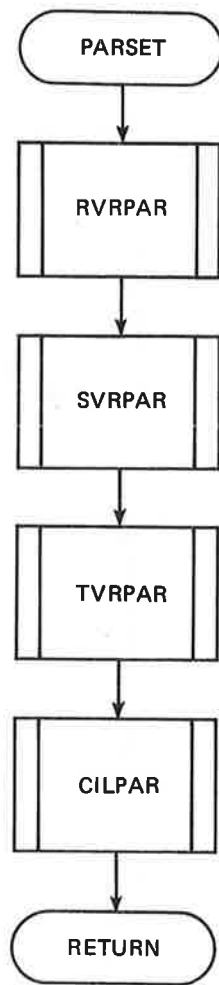


Figure 3-36. Routine to set basic system parameters.

RVRPAR (see Figure 3-37) parameters relating to a transmissometer are input through RVRPAR. These parameters are:

- SCALE: The scale for fixed-point arithmetic associated with the evaluation of RVR is equal to  $2^{\text{SCALE}}$ .
- EXIT: Exit criterion for the literature algorithms of Allard's Law is to exit if the change in RVR is less than  $\text{RVR}/2^{\text{EXIT}}$ .
- VINIT: Initial guess of RVR in the iterative routines.
- TW: The integration time window (in seconds) during which the transmissometer pulse count is accumulated.
- NRWAY: The total number of runways with transmissometer installations.

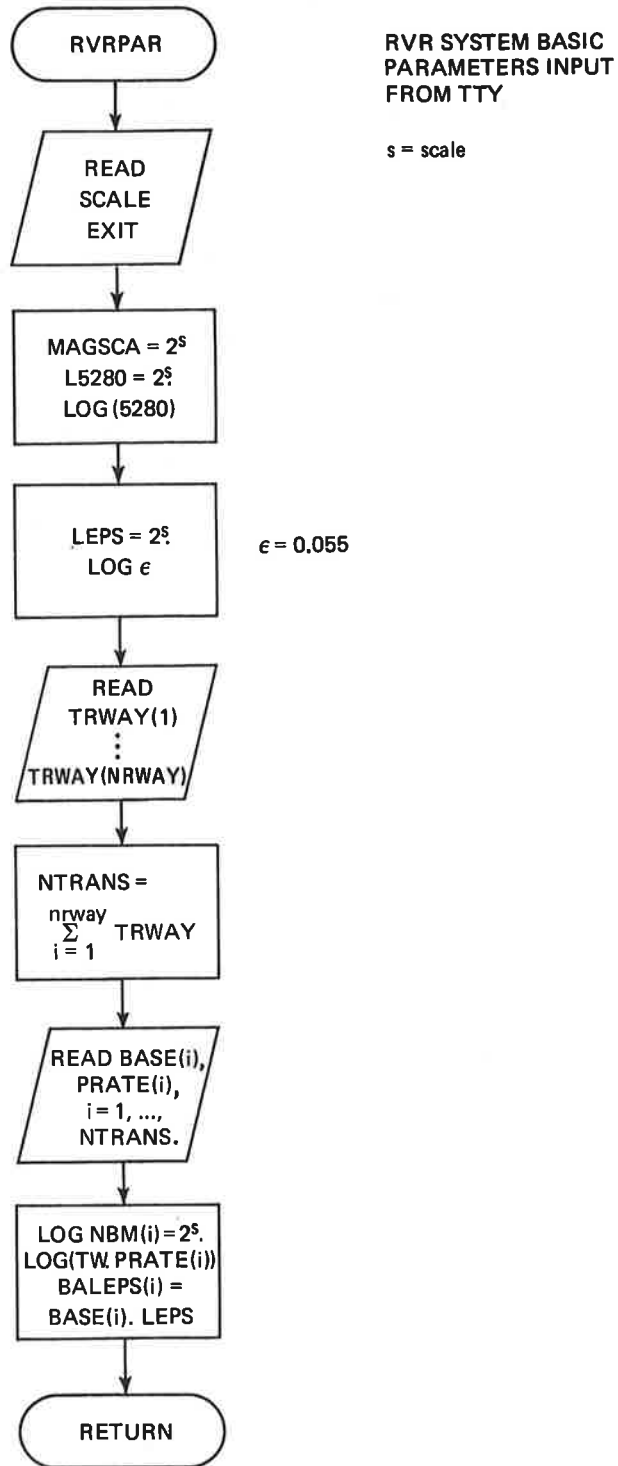


Figure 3-37. Routine to input RVR system parameters.

TRWAY(i), i = 1, --NRWAY: Number of transmissometers per runway.  
The total number of transmissometers NTRANS is evaluated from TRWAY(i).

BASE(i), i = 1, --NTRANS: Base length of transmissometer i.

PRATE(i), i = 1, --NTRANS: Maximum pulse rate of transmissometer i.

SVRPAR, TVRPAR, CILPAR (see Figures 3-38 through 3-40). Parameters relating to SVR, TVR, and the ceilometer measurement system are set up in these routines. These particular systems remain in the conceptual or development stage, so no further elaboration is possible.

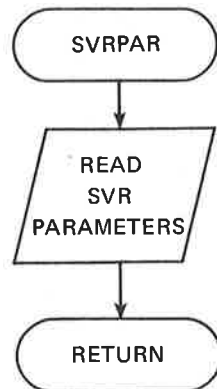


Figure 3-38. Routine to input SVR system parameters.

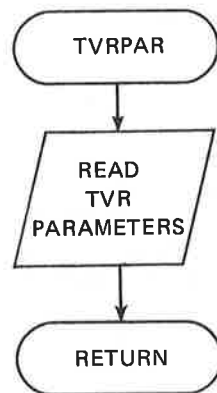


Figure 3-39. Routine to input TVR system parameters.

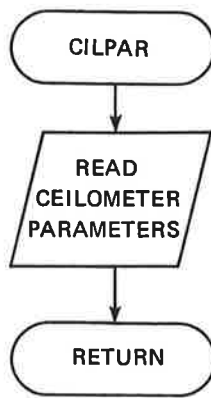


Figure 3-40. Routine to input ceilometer parameters.

### 3.6 INITIALIZATION

Some software and hardware initialization is necessary when the software is first loaded into the memory. The first instruction of the software module START (see Figure 3-41) forms the starting address for the software modules loaded into the memory. The first instruction transfers control to the module INIT (see Figure 3-42) which handles the requisite initializations. Control is subsequently passed to VISIB and it remains there until an alternative operative command is entered from the teletype (after first entering into the monitor mode).

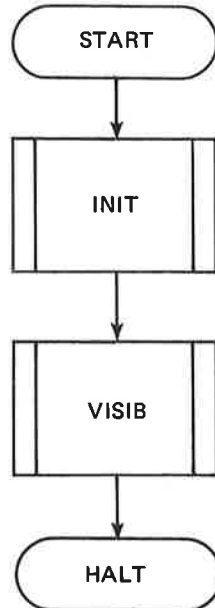


Figure 3-41. Starting routine.

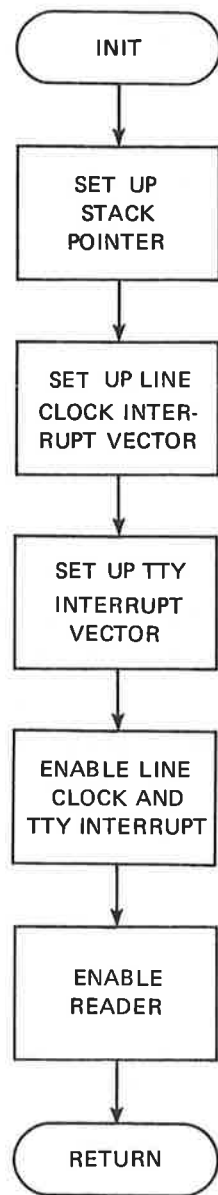


Figure 3-42. Initialization routine.

INIT handles initialization tasks in software and hardware. The following include some of those tasks:

- (1) Software initializations, such as setting up stack pointers, setting up interrupt servicing routine vectors to handle interrupt requests from the real-time clock, I/O devices such as the teletype, data logging systems, etc.
- (2) Hardware initializations, such as enabling the interrupt capability associated with external devices.



## 4. MINICOMPUTER CONFIGURATION

### 4.1 INTRODUCTION

At the core of the data acquisition and computation systems described in this appendix is the minicomputer. Characteristics of the latter used in a repetitive data acquisition and control mode include minimum configuration, low cost, and a fixed storage program. These characteristics are also descriptive of the upper end of the new class of microprocessor, or microcomputer, now generally available.

Two specific configurations are discussed briefly in this section. One is that configuration chosen to implement the simulator-interface described elsewhere in this appendix, and the other represents a desirable specification of the configuration required to implement the field installation.

### 4.2 SIMULATOR-INTERFACE MINICOMPUTER CONFIGURATION

The minicomputer configuration chosen to implement the simulator-interface was required to meet certain criteria not necessarily relevant to the eventual data acquisition and control application. The most important of these was the requirement for powerful software development tools to reduce programming cost to a minimum. It was desired to acquire a minimum-configuration minicomputer system to implement the demonstration control and data acquisition task, and while such a system is consistent with the eventual application, it is not consistent with the necessary initial task of software development. The most cost-effective way of achieving the latter is with a configuration providing maximum support of fast and powerful peripheral equipment including a disc system, line printer, and console keyboard-display.

Initial wordlength versus trade-off studies performed using masking techniques on a large-scale Digital Equipment Corporation (DEC) PDP-10 facility at DOT-TSC determined the need for a 16-bit wordlength computer. Since a cross-assembler and simulator could be accessed on

the PDP-10 facility for development of PDP-11 software, the latter computer was considered. In addition, a PDP-11/15 disc system was available for software development. These considerations were paramount in the decision to acquire a PDP-11/10-AC minicomputer system for the experimental simulator-interface assembly. The configuration consisted of the following components:

- (1) KD11B CPU.
- (2) 8k 16-bit read/write core memory (900 nanoseconds).
- (3) Power fail and restart module.
- (4) Bootstrap loader module.
- (5) Real-time clock, line frequency.
- (6) Teletype interface, line frequency.
- (7) Programmer's console.
- (8) 5-1/4-inch mounting box and power supply.
- (9) ASR-33 teletype and slow-speed paper-tape reader/punch.
- (10) Knee-standing cabinet.
- (11) Software modules.

The photographs of the simulator-interface and the PDP-11/10-AC minicomputer configuration, including the ASR-33 teletype and slow-speed paper-tape reader/punch, are included in Appendix E.

All software was developed using the PDP-10 and PDP-11/15 disc-system facilities. The assembled program was punched on paper tape for subsequent input to the experimental assembly using the slow-speed reader attached to the teletype. Program input using this technique takes approximately 10 minutes.

The choice of the DEC PDP-11 computer to implement the experimental system should in no way imply a preference for this specific computer in the intended eventual application. As noted previously, software development considerations were paramount in the decision. However, similar considerations will require that one of the eventually deployed systems, or a separate facility, incorporate an above-average minicomputer configuration. This will be necessary for software development and maintenance. It is recommended that it include a disc system and line printer as well as the necessary paper-tape peripheral equipment. The cost of this one-only facility will be small compared to the total deployed-system cost, but must be considered an essential part of the system.

Descriptions of the PDP-11 software developed for the experimental system appear in other sections of this report. A brief description and a listing of the assembly language program appears in Appendix D. Also included in that appendix are sections on an estimate of the processor time requirements for an RVR update and on the program size requirements. Upper limits on the processor time of 0.1 second for integer arithmetic and 0.3 second for the floating-point arithmetic calculations seem to be acceptable for a simultaneous update of RVR based on inputs from  $n$  transmissometers. Thus, for a three-runway configuration with three transmissometers per runway, the upper limits on the required processor time are 0.9 second and 2.7 seconds, respectively, for integer and floating-point arithmetic RVR calculations. The time estimates are based on a minimal instruction set which does not include a hardware multiply and divide capability. With the present trend of lower costs of fabrication, hardware capability of multiply and divide for both fixed as well as floating-point arithmetic is likely to be available, if so desired, with only a small addition to the total cost of the SDCU configuration. The total size of the simulator software is 1754 memory locations. Thus, it can be inferred that the RVR update processing and associated I/O operations can be carried out with adequate speed and accuracy on a 16-bit minicomputer with 4k memory and memory cycle times of about 1.2 seconds. It is not possible to define the time and size requirements for other visibility system parameters such as SVR, TVR, and ceiling, until the associated systems and calculations are well defined.

#### 4.3 MINICOMPUTER SPECIFICATION

The investigation described in this appendix provides a basis for the specification of a minicomputer configuration to implement the evolutionary system. The specification is written integrally with the interface specification, since the two components are interdependent. The computer specification can be easily met by many low-cost minicomputers currently in the market-place. The dynamic nature of the technology should ensure that even lower-cost mini- or microcomputers will meet the specification in the near future. The specification is not part of this appendix and appears under separate cover.

## 5. INTERFACE CONFIGURATION AND DESIGN

### 5.1 INTRODUCTION

This section addresses the topic of communication between external devices and sensors and the minicomputer. The general configuration of an interface to accomplish this task in the current application is delineated and its modular character examined.

The design and mechanization of a simulator-interface is described whereby simulated sensor inputs are interfaced with the software-minicomputer system to verify the conceptual system design and to identify any special hardware/software problems.

### 5.2 INTERFACE CONFIGURATION

Communication between field-located sensors and the computer and between the computer displays, controls, and switches, takes place through an interface.

The interface consists of registers that receive or transmit information via a bus to the computer. These registers may be either flip-flop storage registers or dynamic signals which are simply gated to the bus during a transfer. In general, registers are both loadable and readable from the bus, but special-purpose interfaces usually contain one-sided registers, that is, either "read only" or "write-only". To maintain the transmission-line characteristics of the bus, special circuits are required to pass signals to and from the bus. The majority of bus signals are received, driven and terminated as shown in Figure 5-1. Information is received from the bus using gates which have a high input impedance and proper logic thresholds. Information transmitted on the bus must be driven with open collector drivers capable of sinking a specified current with a specified collector voltage and with less than a specified output leakage current.

Interface configuration is essentially similar for all contemporary minicomputers, but the detail design and specification is

minicomputer-specific. A block diagram of a typical interface for the specific application addressed in this appendix is shown in Figure 5-2. Registers to receive and transmit information are shown for typical parallel and serial I/O data contemplated in this application. Thus, binary counters are shown to receive pulse-rate information, for example, from transmissometers and possibly the ceilometer and slant-visual-range instrumentation. Other registers receive discrete-type information from switches or relays. An example of the latter would be the approach or high-intensity-light-switch setting. Often only a few bits of a 16-bit word will be necessary to transmit information about a discrete switch setting, so that, in general, a single-word register can contain information from more than one source. This approach should be followed whenever possible to economize on interface hardware. Some minicomputer types provide for discrete-bit I/O to simplify control of contacts, relays, etc. Software coding and decoding is relied upon to establish control or status of the devices.

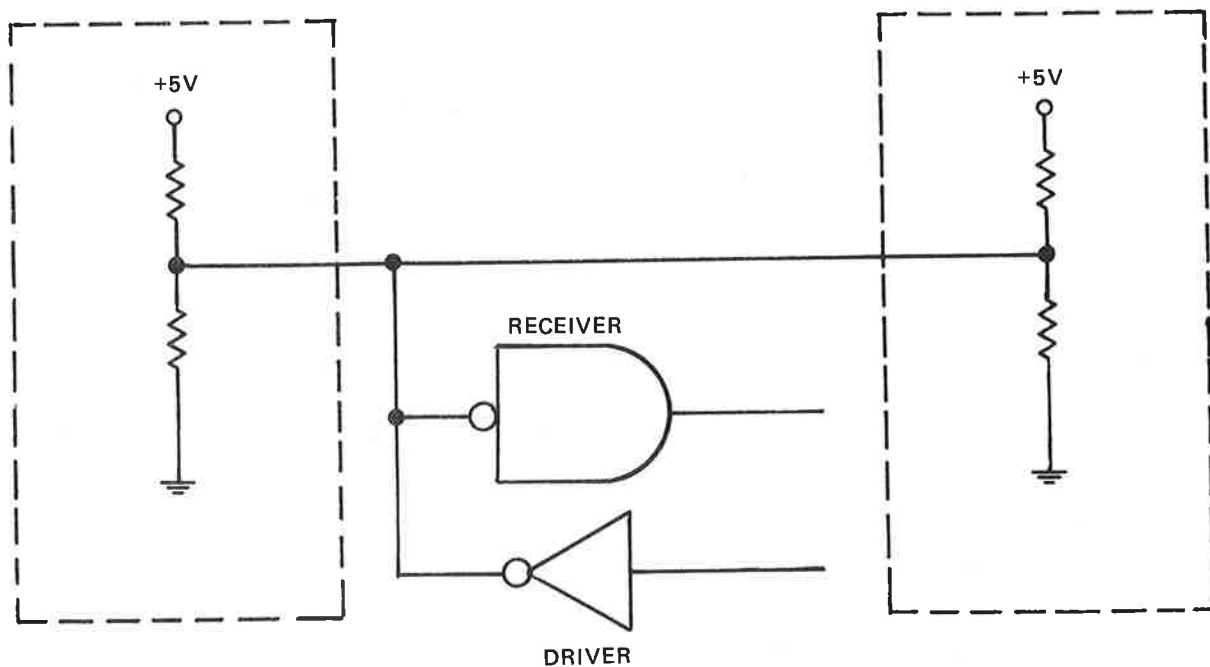


Figure 5-1. Typical bus line.

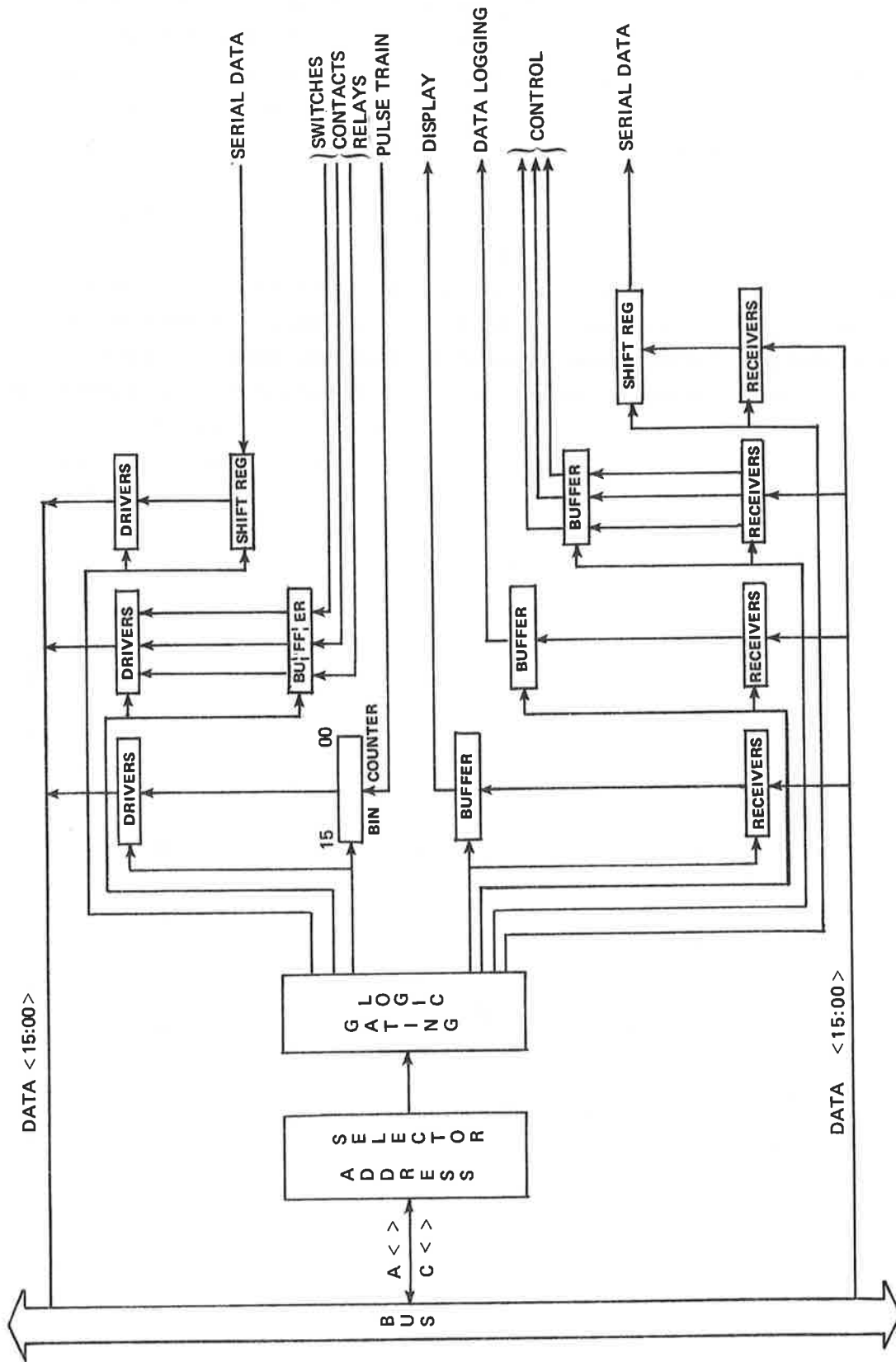


Figure 5-2. Typical signal data converter I/O interface configuration.

Output registers store specific information strobed from the computer. The information may be intended to display, for example, the runway visual range, slant-visual range and ceiling or intended to control external devices such as the intensity of the runway lights or to sound an alarm when a specific event takes place. Data logging is another output function. As before, a full 16-bit word may contain information intended for more than one destination. Definition and specification of the output interface essentially ends at the register. Local displays and controls are designed to receive their information as parallel data directly from these registers. Transmission of register information to distant locations (for example, remote displays) requires a serial data transmission. Some minicomputers have serial data I/O channels, in addition to the common parallel channels, to facilitate the transmission of data to or from remote devices. In general, however, the task of converting data from parallel- to serial-bit format and providing an appropriate gated clock signal for serial I/O operations is considered part of the interface function.

Associated with each interface or device register is a unique address by which communication with the computer is possible. When information in a particular register is required by the computer, the address of that register is put on the bus and decoded by another major interface element, the address selector. The latter transmits control signals to the addressed register which gates the requested information onto the bus via the data lines. Similarly, information is transferred from the computer to specified interface output registers.

### 5.3 MODULAR INTERFACE DESIGN

From the foregoing description, it is possible to see the potential for modular interface design. A field installation would require multiple combinations of the five types of registers discussed in Subsection 5.2, namely binary counters to register pulse-rate data, parallel-word input registers containing discrete switch, contact, or relay data, parallel-word output registers to transmit local display, data logging or control information, and shift registers to transmit and receive serial data; the latter required only for communication with remote devices.

Each of the five types of I/O interface would contain an associated address selector for address decoding and either receiver or driver networks. Each would contain a convenient number of (identical)

registers or words, probably two, three, or four, depending upon the interface type. For example, three would be a convenient number for the pulse-rate input module, since three transmissometers will be associated with each runway (at touchdown, midpoint, and rollout locations). The I/O interface cards would plug into a common, multi-slot chassis with common connection to the bus.

A modular approach to interface design is recommended so that the size and capability of a typical system can be matched to the particular requirements of the airport installation. It is consistent with the modular character of the minicomputer and software design. If airport requirements expand, the system can grow to meet the needs with the addition of the appropriate modules in either the interface or minicomputer or both.

#### 5.4 RVR SIMULATOR-INTERFACE DESIGN

The design of the RVR simulator-interface is instructive because it contains examples of all of the register types that will be required in the field installation. The experimental interface is designed to accept inputs from simulated sensors and transmit this data, upon request, to the minicomputer. An output register accepts data from the computer for subsequent visual display of RVR. The simulated sensors produce signals of the type specified for actual field-installed sensors. They include pulse-train signals of the type and bandwidth required from transmissometers and also possibly the ceilometer and slant-visual-range instrumentation; signals from discrete switch settings or relays representing signals from the runway high-intensity light setting switch or background luminance sensor or runway selector; binary-coded signals from a controller-selected decimal panel switch representing an "alert" RVR value or from a serial-loaded shift register representing the interface for a serial transmission from a remote sensor. All I/O signal levels are consistent with interface logic, namely, 5 volts.

A block diagram of the simulator-interface is shown in Figure 5-3. A detailed schematic of the simulator-interface is shown in Figure 5-4, and views of its three components are shown in Figures 5-5 through 5-7. Figure 5-5 shows the front control panel of PDP-11/10-AC minicomputer. Figure 5-6 shows the ASR-33 teletype and slow-speed paper-tape reader/punch. Figure 5-7 shows the front panel of the simulator-interface showing the switches controlling various simulated-sensor inputs and the RVR digital display. The function of the switches is briefly described below. A ten-position switch for transmissometer



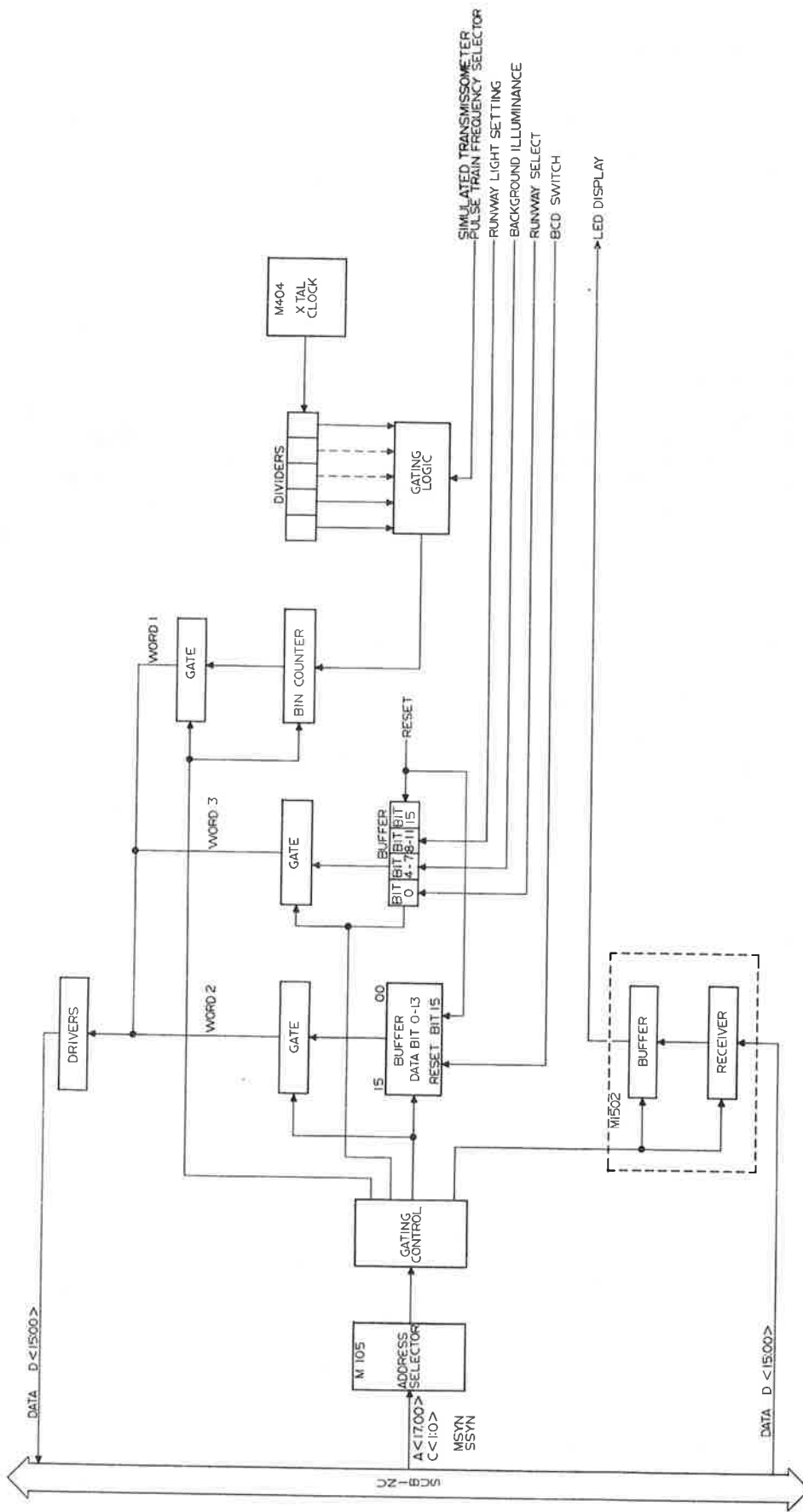


Figure 5-3. Block diagram of the simulator-interface.

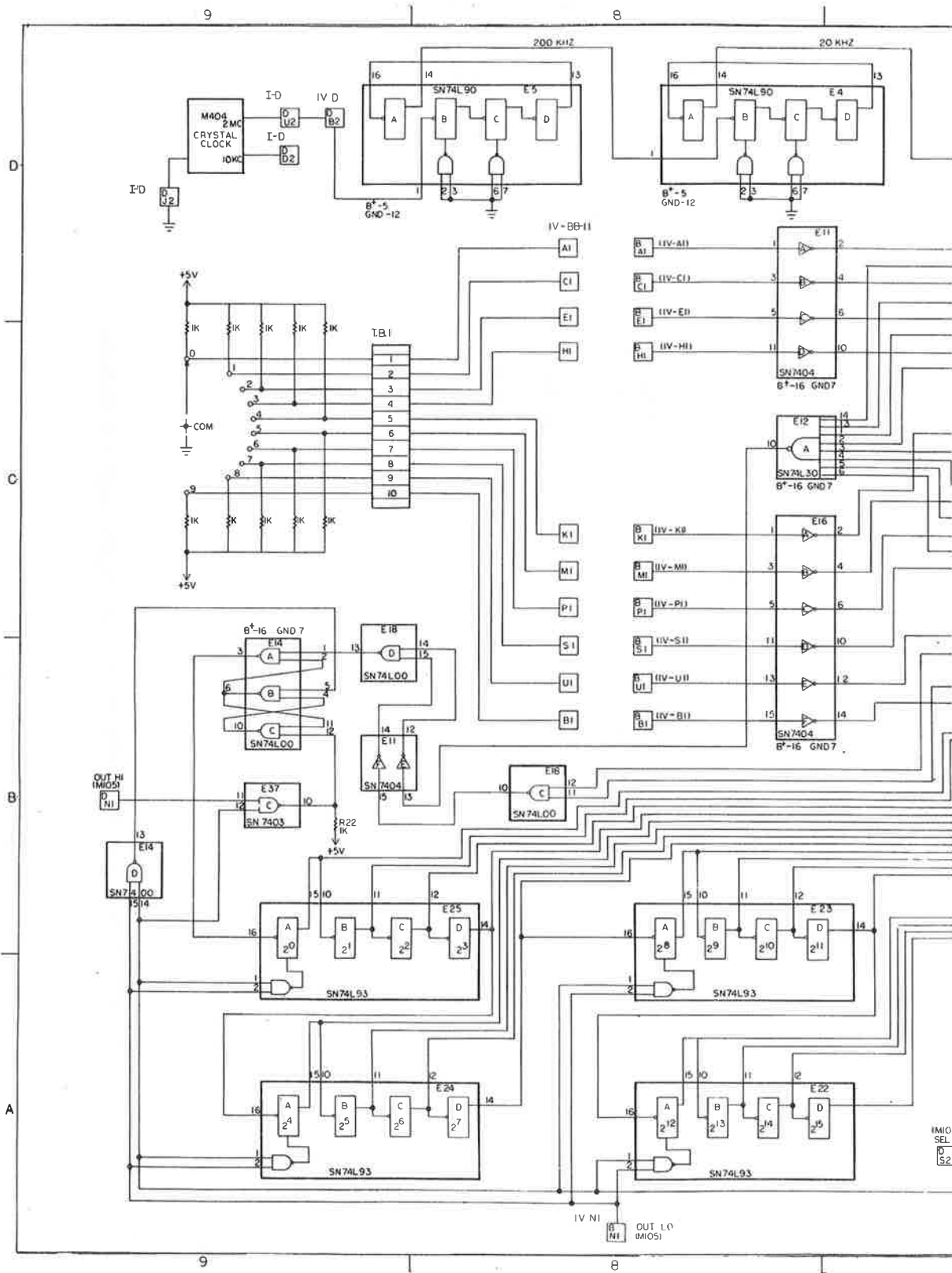


Figure 5-4. Detailed schematic of the simulator-interface (page 1 of 4).

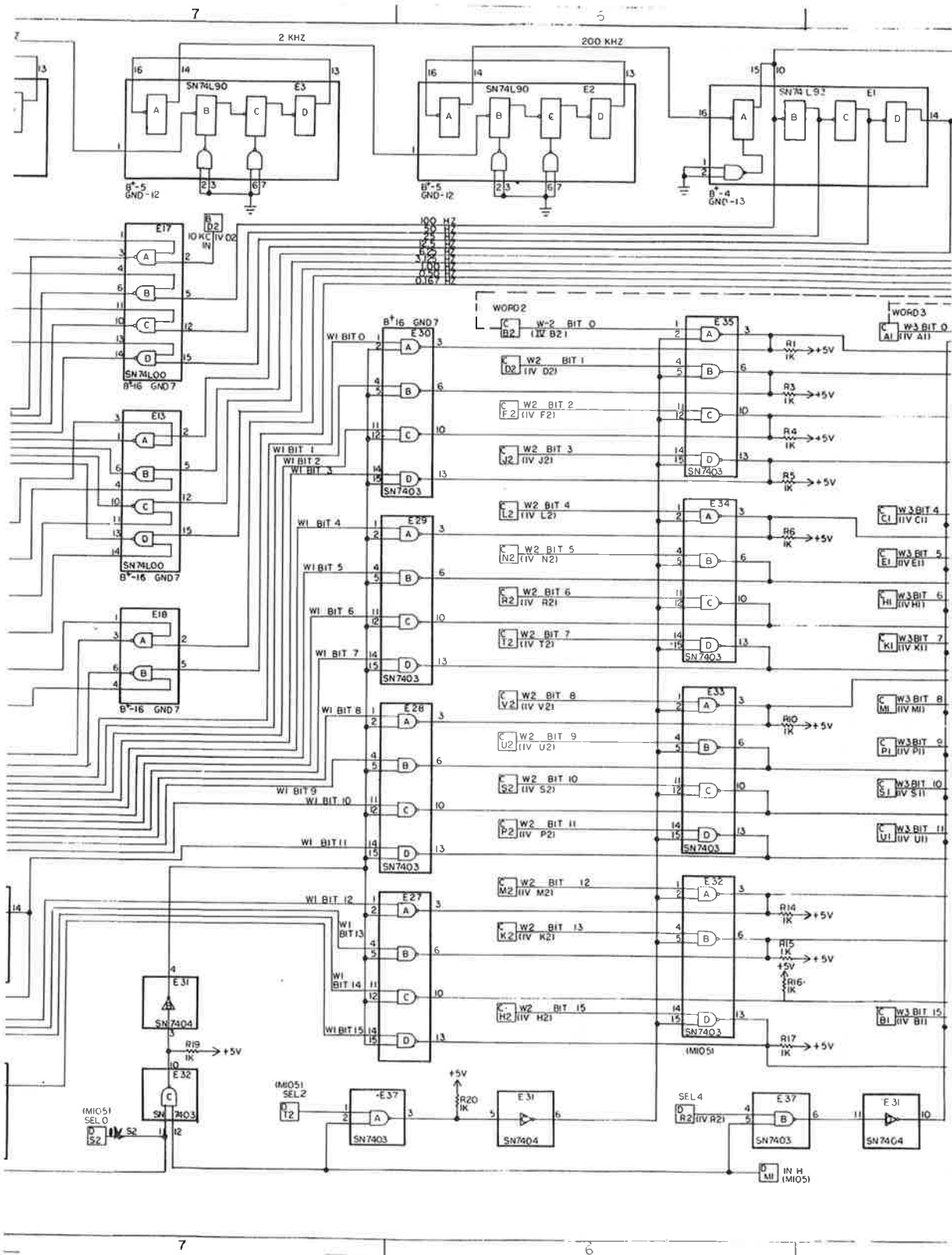


Figure 5-4. Detailed schematic of the simulator-interface (page 2 of 4).

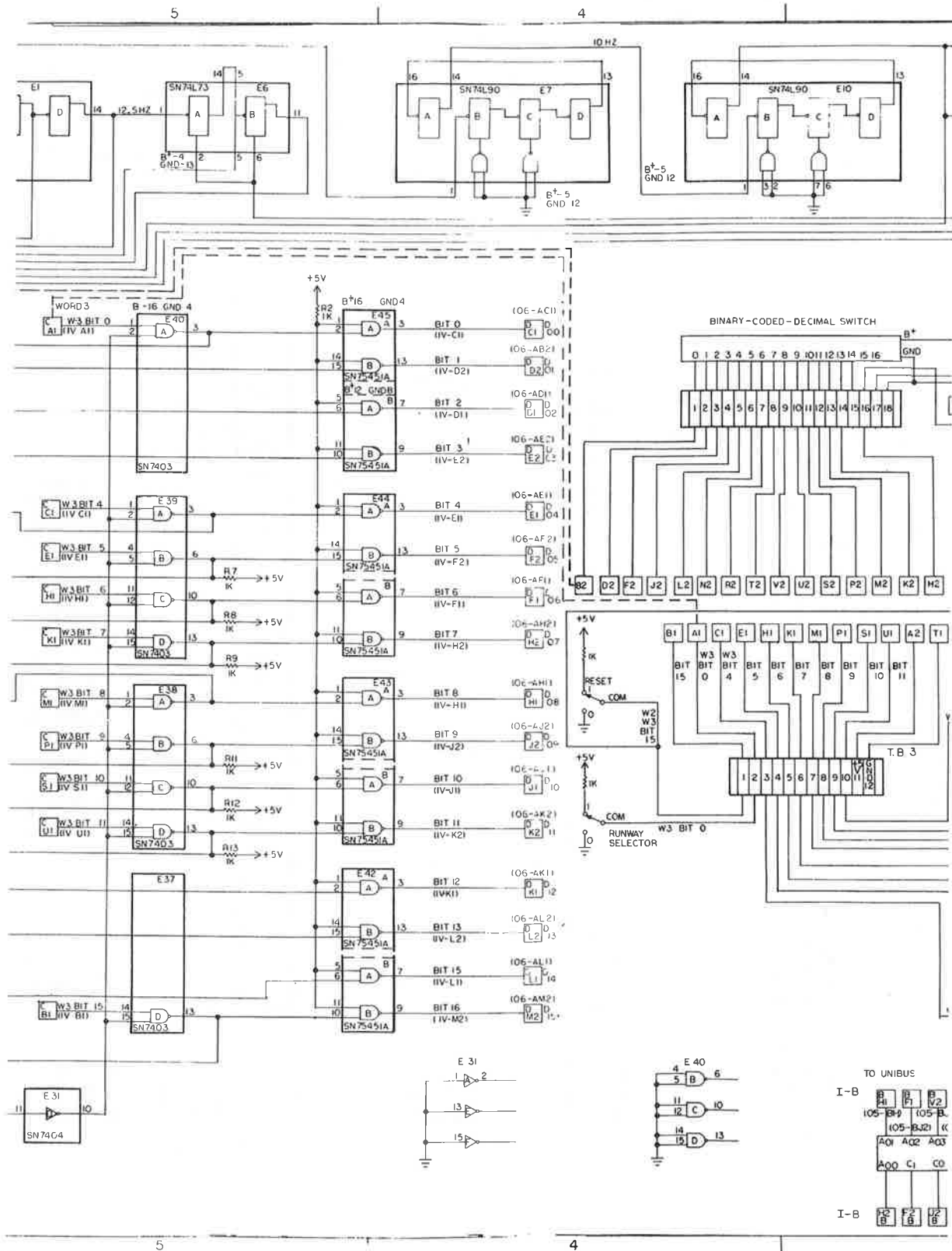


Figure 5-4. Detailed schematic of the simulator-interface (page 3 of 4).

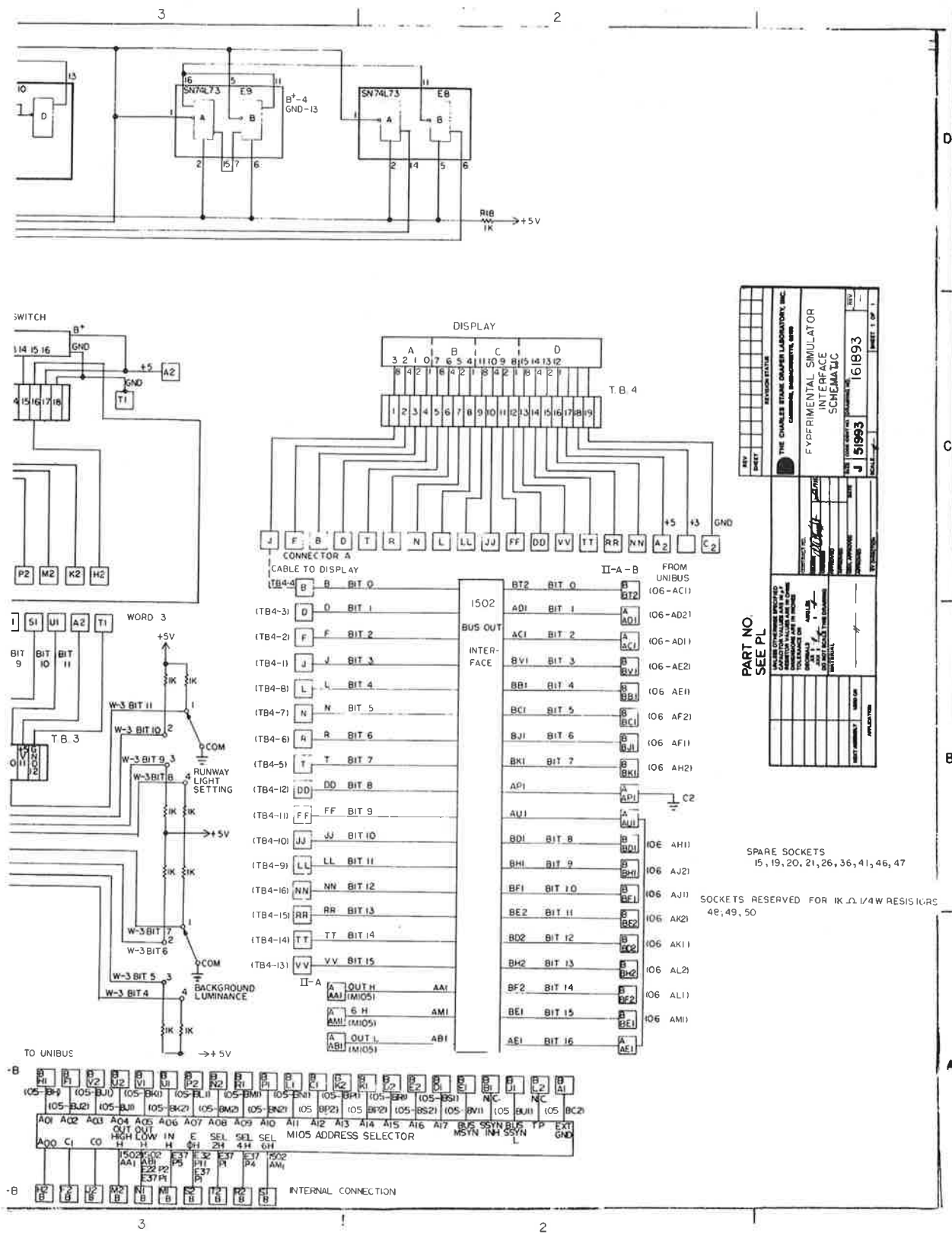


Figure 5-4. Detailed schematic of the simulator-interface (page 4 of 4).

No. 1 provides a capability of choosing one of the ten pulse rates ranging from 0.167 to 10,000 pulses per second to simulate the pulse rate mode of transmission. A four-digit position switch inputs a binary-coded decimal count to the computer to simulate a serial-loaded register representing interface for a serial transmission. A four-position switch simulates the discrete information on the runway light intensity. Similarly, a four-level discrete information characterizing the background luminance is provided by a four-position switch. A two-position reset switch provides capability to change the various switch settings appropriately during a software RVR update cycle. Further details on the function of the reset switch and the information represented by different positions of the switches described above can be found in Appendix D under the section on software description. RVR for the transmissometer selected by the runway-selector switch is shown on the visual LED display either every time new RVR is calculated for that transmissometer or the runway-selector switch position is changed. The latter action does not interfere with the software calculations that may be in progress.

#### 5.4.1 Word 1: Discrete-Frequency Pulse-Train Input

To simulate inputs from transmissometers or other remote sensors, a crystal clock and frequency dividers were used to generate ten discrete-frequency pulse-trains. The discrettes were originally chosen to span the frequency range from zero, corresponding to zero transmittance, to 200 Hz, corresponding to 100% transmittance. Subsequently, a decision was made to expand the bandwidth of the transmitted signal to 10 kHz. Consequently, the highest discrete frequency was changed from 200 Hz to 10 kHz. The discrettes implemented are shown in Table 5-1 with the associated decimal switch code.

The switch gates the appropriate frequency pulse-train to the binary-counter flip-flops. The pulse count is incremented in the register until software requests the current value. The count is first inhibited, then the static contents of the register strobed onto the bus. The register is reset before a new count begins. Bits 0 to 15 of word 1 are used.

#### 5.4.2 Word 2: Binary-Coded Decimal Switch

This serial-loaded binary work register simulates the interface for an alternative means of transmitting data from remote sensors,

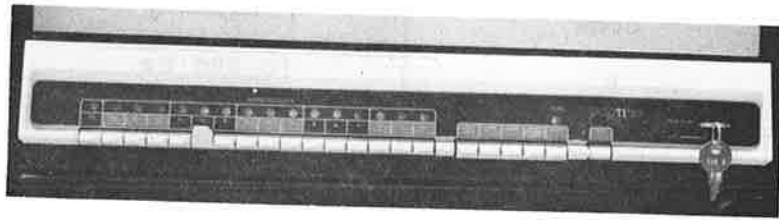


Figure 5-5. Front control panel of the PDP-11/10-AC minicomputer of the simulator-interface minicomputer configuration.



Figure 5-6. ASR-33 teletype and slow-speed paper-tape reader/punch.

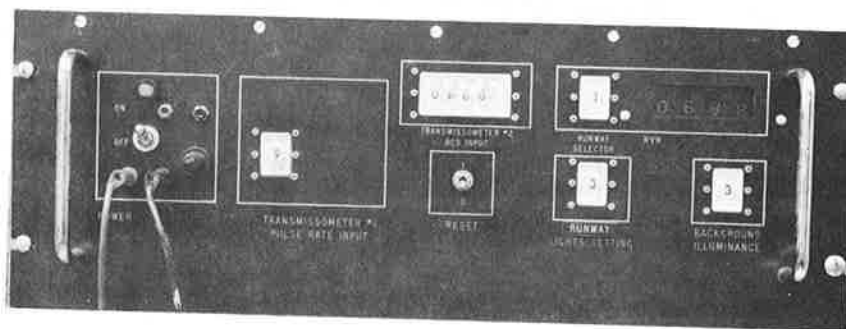


Figure 5-7. Front control panel of the signal simulator bearing switches controlling the simulated sensor inputs to the minicomputer and the visual RVR digital display.

TABLE 5-1. PULSE-TRAIN DISCRETE-FREQUENCY CODE

Code	Frequency
0	10,000 Hz
1	100
2	50
3	25
4	12.5
5	6.25
6	3.125
7	1.000
8	0.500
9	0.167

namely, serial data transmission. This technique requires that the sensor output be converted into digital form, coded, and transmitted serially, either synchronously or asynchronously, to the interface. The register also simulates a decimal console-panel switch into which the controller could load an "alert" RVR value. The decimal value is coded internally into binary form before being gated to the bus. Bits 0 to 15 of word 2 are used.

#### 5.4.3 Word 3: Discrete Switch, Relay, or Control Settings

This register contains information on discrete switch settings or control relay positions that must be communicated to software. The specific functions simulated were the runway-light-intensity setting (bits 8 to 11), the background-luminance setting (bits 4 to 7), the runway selector (bit 0), and the reset switch (bit 15). The latter is unique to the simulator-interface, being reset whenever a change is made to word 1 or 2. Bits 1, 2, 3, 12, 13, and 14 of word 3 are unused.

#### 5.4.4 Word 4: Output Register

This register receives output signals from the computer that are destined for local or remote display, to control switches or relays, or for data logging. The flip-flop register of the simulator-interface drives a local four-digit LED display of RVR. All 16 bits of the word are used.



## 5.5 IMPLEMENTATION OF THE SIMULATOR-INTERFACE

This subsection describes the basic assumptions and definitions for the simulator-interface, and the ground rules under which the combined software/hardware must operate. A description of the mechanization is also given. A schematic of the simulator-interface is shown in Appendix E.

A typical sequence of operations that must occur is indicated in Table 5-2.

The interface contains a DEC M105 Address Selector, an M1502 Bus Output Interface, and an M404 Crystal Clock. All other functions, including input registers, are implemented on a blank wire-wrappable module. This card and the DEC cards plug into a BB11 module connector. The whole subassembly is mounted in a chassis and connected to the computer via a BC11A bus cable. Only 5-volt power is needed for the interface, and this is supplied by an external supply.

An 18-bit address A (17:00) is decoded. A (17:13) must all be ones and set by software. A (12:03) is determined by jumpers on the M105 card. When the jumper is "in", the computer will look for a zero on that address line. The addresses to be used are 764000 to 767777. So if addresses 764000, 764002, 764004, and 764006 are to be used, jumpers are placed in bits 3, 4, 5, 6, 7, 8, 9, 10, and 12. The M105 controls up to four addresses. A00, A01, A02, C00, and C01 are used for control.

Note that the address supplied on the bus is that of the even-numbered location and the next higher odd location is selected as well. A02 and A01 provide a coding array for the four selected addresses. A00 is for byte operation. A DATI to 764000 transfers data to locations 764000 and 764001 in two bytes. A DATOB to location 764006 loads only location 764006.

It is important to bear in mind that the sequence of operations is critical; otherwise the data may be lost or will be in error. Referring to the chart, assume that the computer is ready to start a read/write sequence. Step 1 must occur first. This step clears the 16-bit binary counter and sets a clock-enable flip-flop. This allows the 16-bit counter to be clocked at the selected rate determined by the ten-position selector switch. The period of counting is now set by the computer, and must be small enough so the counter will not overflow. Immediately after this predetermined period, step 2 must

TABLE 5-2. TYPICAL SEQUENCE OF I/O OPERATIONS FOR THE SIMULATOR-INTERFACE

STEP NO.	OPERATION	ADDRESS BITS				COMMENTS
		A02	A01	A00	C00	
1	Initialize—Clears 16-bit counter and sets clock-enable flip-flop	0	0	0	1	DATOB—Sets Select 0 and Out Low to ones Data bits—not applicable Bus address = 764000
2	Resets clock-enable flip-flop, prevents further counting	0	0	1	1	DATOB—Sets Select 0 and Out Hi to ones Data bits—not applicable Bus address = 764001
3	Reads transmissometer counter into computer	0	0	0	0	DATI—Sets Select 0 and INH to ones Strobes word 1 into computer Bus address = 764000
4	Reads preset transmissometer data into computer	0	1	0	0	DATI—Sets Select 2 and INH to ones Strobe word 2 into computer Bus address = 764002
5	Reads word 3 into computer	1	0	0	0	DATI—Sets Select 4 and INH to ones Strobes word 3 into computer Bus address = 764004
6A	Output runway visual range to display	1	1	0	1	DATOB—Sets Select 6 and Out Low to ones Puts Low Order byte to display Bus address = 764006
6B	Output runway visual range to display	1	1	1	1	DATOB—Sets Select 6 and Out Hi to ones Puts High Order byte to display Bus address = 764007

occur. This will clear the clock-enable flip-flop inhibiting any further counting. The binary counter, however, will retain its last value until it is cleared again by repeating step 1. Step 1 should not occur until the data is read into the computer.

After step 2 is performed, steps 3, 4, and 5 may be performed in any sequence. The three words are now stored in the computer. Steps 6A and 6B should be done sequentially. Since the data to the display is output in two bytes, the time delay between the two DATOB output instructions will be small enough so that the eye will not see it.

Word 3 does not use all 16 bits. Bit 0 is either a 1 or 0, and is set by the runway-selector switch. Bits 1, 2, and 3 are not used. Bits 4, 5, 6, and 7 are the output of the four-position background-luminance switch. Only one of the four bits will be enabled at any time. Bits 8, 9, 10, and 11 are the outputs of the four-position runway-light-intensity setting switch. Again, only one of the four bits will be enabled at any one time. Bits 12, 13, and 14 are not used. Bit 15 is the output of a reset switch. This reset switch is set while changing the front panel switches such as the BCD selector switch or the transmissometer frequency selector switch.

It is apparent from a study of the simulator-interface schematic in Appendix E that a common set of drivers is employed for words 1, 2, and 3. This is possible because only one word at a time can be addressed and driven onto the bus. Control signals from the address selector gate the appropriate word to the drivers. Similarly, this approach would be followed in the design of the modular interface for the field installation. Each individual multi-word interface card would employ a common set of drivers or receivers.

## 5.6 CHARACTERIZATION OF THE INTERFACE SENSOR INPUTS

Previous sections of this subsection have described the configuration of an interface to communicate between sensors or devices and the computer. It is clear that the interface-software system is very flexible so that a variety of alternative input signal formats can be considered as candidate specifications. This flexibility is important, indeed it is a prime characteristic of such a hardware/software system, in that it provides freedom for the unconstrained design of new sensors and modification of existing sensors to take full advantage of new technology. This is also true in the selection of sensor data transmission technique.

The modular construction envisioned for the field-interface installation provides for a multiplicity of I/O signal formats, thus the selection of a format for a particular sensor will not be constrained by interface design. Consideration of the type of information the signal conveys and its time domain characteristics, as well as transmission technology considerations, will be determining factors.

Input signals to current SDC systems and anticipated additional inputs to future visibility-monitoring configurations are reviewed in Subsection 5.6 with respect to their time domain, signal format, and transmission characteristics.

#### 5.6.1 Transmittance Measurement Sensors

RVR transmissometers, background-luminance sensors, anticipated TVR, SVR, and ceilometer sensors all fall into this category. The signals from these sensors are characterized by an inherent spatial integration. Further processing to provide a useful visibility measurement includes temporal integration. Currently, the transmission from remote transmissometers has a pulse-train signal form, the pulse rate being a measure of transmittance. The temporal integration for these signals simply involves a count of the pulses over a given period of time, a function which can be easily and simply provided by suitable hardware counting logic located in the SDC I/O interface. The pulse counting can be accomplished by the computer software also, however at an expense in the form of increased computational time and memory size requirements. Moreover, the increased cost of the necessary hardware logic modules to interrupt the software program on the arrival of a new pulse more than offsets the savings that result from having dispensed with the hardware counters. These considerations tend to rule out the pulse counting by the software when the sensor data is transmitted using pulse rate modulation. The 20-meter baseline transmissometer under development at TSC uses a similar signal format but with bandwidth increased to 10 kHz. Advanced semiconductor technology drivers transmit pulses up to 10 MHz over very long distances using a twisted-pair line. The latter is terminated with a receiver that presents the pulse train at proper logic levels to the interface. It is recommended that the background-luminance sensor under development at TSC and TVR sensors transmit data in a similar manner to the 20-meter base transmissometer, using a common bandwidth for the transmissivity measurement.

SVR and ceilometer instrumentation is currently in the experimental stage. However, the extent of the calculations, including deconvolution, known to be involved in the reduction of a measurement of backscattering from a laser pulse, would appear to preclude the use of the central minicomputer for this data processing. Both analog and digital preprocessing of the backscattering data require further investigation.

Digital processing has the advantage in that a dedicated minicomputer could handle data processing and instrument control. The signal format and transmission of a transmittance measurement from either the SVR or ceilometer sensors need not be specified at this time. Commonality with other transmittance sensors would suggest a similar pulse-train signal format. However, the discrete measurement nature of the new sensor might make a BCS (or similar) serial transmission more convenient. The cost or other impact on interface design is negligible.

#### 5.6.2 Status of Switches and Relays

Relay and switch positions providing the status of high-intensity runway lights, approach lights, or discrete-level background luminance appear as discrete (static) logic levels at the interface. This is a straightforward procedure for local (parallel) transmission of data, as was demonstrated on the design of the simulator-interface. Only two lines are necessary to transmit information from a four-position switch; three lines from an eight-position switch. When switches or relays are remotely located, a serial data transmission may be necessary.

Data must be conditioned by appropriate line drivers and receivers, presenting the data at correct logic levels for loading into the shift-register on the appropriate interface card. Gated clock signals must be provided by the minicomputer or an external reference. Serial data transmission is typically at a pulse rate of 100 to 1000 kHz.

An alternative method of transmitting remote switch and relay data is by frequency multiplexing. This technique involves transmitting a discrete frequency pulse-train corresponding to each switch, relay, or control position. The interface to accept this data is identical to that provided for pulse rate information from the transmissometers. Appropriate line drivers and receivers would be required as with all remote data transmission. The design of the discrete frequency pulse-train transmitter would be similar to the pulse-train simulator detailed in a previous section.

## 5.7 INTERFACE SPECIFICATION

The detailed design of an interface assembly depends upon the specific minicomputer choice. For this reason, specification of the interface has been made integral with that of the minicomputer to implement the eventual evolutionary system (see Subsection 4.3).

## APPENDIX A

### ITERATIVE ALGORITHMS FOR RVR COMPUTATION

#### A.1 INTRODUCTION

An investigation is made of the performance of different iterative algorithms to solve a nonlinear equation  $y = f(x)$  for its zero. All iterative algorithms considered are single point in character and utilize the information on the value of the function,  $f(x)$ , and its higher derivatives at a given point to arrive at the next guess. First, the Newton-Raphson method is investigated, which utilizes the value of the function and its slope, and then three different methods are considered, all utilizing higher-order information up to second derivatives of the function.

Let  $\alpha$  be the zero of  $y = f(x)$ , i.e.,  $f(\alpha) = 0$ .

#### A.2 NEWTON-RAPHSON METHOD

The Taylor series expansion of  $y = f(x)$  about point  $x_i$  is given by

$$y = f(x_i) + (x - x_i)f'(x_i) + \frac{1}{2}(x - x_i)^2 f''(p), \quad x < p < x_i \quad (\text{A-1})$$

Since  $f(\alpha) = 0$ , we have

$$0 = f_i + (\alpha - x_i)f_i' + \frac{1}{2}(\alpha - x_i)^2 f''(p), \quad (\text{A-2})$$

dropping the arguments.

Equation (A-2) suggests the following iteration formula:

$$x_{i+1} = x_i - \frac{f_i}{f_i'}, \quad (\text{A-3})$$

which is the familiar Newton-Raphson method. Substituting for  $x_i$  in (A-2) and using (A-3),

$$0 = f_i + \left( \alpha - x_{i+1} - \frac{f_i}{f_i'} \right) f_i' + \frac{1}{2} (\alpha - x_i)^2 f''(p),$$

or

$$\epsilon_{i+1} = -\frac{1}{2} \frac{f''(p)}{f_i''} \epsilon_i^2 \quad (\text{A-4})$$

where  $\epsilon_i = \alpha - x_i$  is the error at stage  $i$ . Thus, Newton-Raphson has a second order of convergence.

### A.3 HIGHER-ORDER METHODS UTILIZING SECOND DERIVATIVE

Let  $y = f(x)$  have an inverse  $x = g(y)$  in the neighborhood of a root  $\alpha$  of  $f(x) = 0$ . The Taylor series expansion of  $g(y)$  about a point  $y_i$  is given by

$$x = g(y) = g(y_i) + (y - y_i)g'(y_i) + \frac{1}{2}(y - y_i)^2 g''(y_i) + \frac{1}{6}(y - y_i)^3 g'''(p), \quad (\text{A-5})$$

where  $p$  is between  $y$  and  $y_i$ . Since  $\alpha = g(0)$ , we have:

$$\alpha = x_i - f_i g_i' + \frac{1}{2} f_i^2 g_i'' - \frac{1}{6} f_i^3 g_i'''(p), \quad (\text{A-6})$$

where  $y_i = f(x_i) = f_i$ .

It is easy to deduce from the inverse relationship of the functions  $y = f(x)$  and  $x = g(y)$ , that:

$$f_i' g_i' = 1. \quad (\text{A-7})$$

By successively differentiating (A-7), we evaluate:

$$g_i'' = \frac{-f_i''}{(f_i')^3} \quad (\text{A-8})$$

and

$$g_i''' = \left( \frac{-1}{(f_i')^3} \right) \left( \frac{f_i'''}{f_i'} \right) - \frac{3f_i''^2}{(f_i')^2}. \quad (\text{A-9})$$



#### A.4 METHOD 2

Equation (A-6) suggests the following iteration formula:

$$x_{i+1} = x_i - \frac{f_i}{f_i'} \left( 1 + \frac{1}{2} \frac{f_i}{f_i'} \cdot \frac{f_i''}{f_i'} \right), \quad (\text{A-10})$$

which is the same as what has been referred to as Method 2.

Subtracting (A-10) from (A-6), we have the following expression for error at each iteration:

$$\epsilon_{i+1} = -\frac{1}{6} g'''(p) f_i^3,$$

since

$$\begin{aligned} f_i &= f(x_i) = f(x_i) - f(\alpha) = (x_i - \alpha) f'(q) \\ &= -\epsilon_i f'(q), \text{ with } q \text{ between } \alpha \text{ and } x_i \end{aligned}$$

$$\epsilon_{i+1} = \left( \frac{1}{6} f'(q) g'''(p) \right) \epsilon_i^3. \quad (\text{A-11})$$

If the root  $\alpha$  is simple, the expression in brackets is bounded in some interval of  $\alpha$ . Thus, the order of the iteration formula (A-10) is three.

#### A.5 HALLEY'S METHOD

Consider the iteration formula

$$x_{i+1} = x_i - f_i g_i' + \frac{1}{2} f_i^2 g_i'' = x_i - u_i (f_i' g_i' - \frac{1}{2} f_i f_i' g_i'').$$

Our object is to approximate (A-1) by the rational formula:

$$x_{i+1} = x_i - \frac{P_0}{1 + Q_1 u_i} \cdot u_i,$$

where

$$u_i = \frac{f_i}{f_i'},$$

and choose  $P_0$  and  $Q_1$ , such that

$$f_i' g_i' + \frac{1}{2} f_i f_i' g_i'' = 1 + \frac{1}{2} u_i \frac{f_i''}{f_i'}$$

and

$$\frac{P_0}{1 + Q_1 u_i}$$

have as many derivatives equal as possible at  $u_i = 0$ . We would like to choose two constants at our disposal,  $P_0$  and  $Q_1$ , such that

$$\left( 1 + \frac{1}{2} u_i \frac{f_i''}{f_i'} - \frac{P_0}{1 + Q_1 u_i} \right),$$

and its first derivative with respect to  $u$  is equal to zero at  $u_i = 0$ . Consider

$$\left( 1 + \frac{1}{2} u_i \frac{f_i''}{f_i'} \right) - \frac{P_0}{1 + Q_1 u_i} = \frac{\sum_{j=0}^{\infty} H_j u_i^j}{1 + Q_1 u_i}.$$

This requirement is met if  $H_0 = H_1 = 0$ . Then:

$$P_0 = 1 \quad \text{and} \quad Q_1 = -\frac{1}{2} \frac{f_i''}{f_i'}$$

Thus, we arrive at Halley's iteration formula

$$x_{i+1} = x_i - u_i \frac{1}{\left( 1 - \frac{1}{2} u_i \frac{f_i''}{f_i'} \right)}. \quad (\text{A-12})$$

Subtracting (A-12) from (A-6), we get the expression for the error:

$$\epsilon_{i+1} = -\frac{1}{6} g'''(p) f_i^3 - \frac{1}{2} \frac{f_i''}{(f_i')^4} \left( \frac{1}{1 - \frac{1}{2} u_i \frac{f_i''}{f_i'}} \right) \cdot f_i^3.$$

Now  $f_i = -\epsilon_i f'(q)$  with  $q$  between  $\alpha$  and  $x_i$  as explained earlier. Thus,

$$\epsilon_{i+1} = f'(q) \left[ \frac{1}{6} g'''(p) + \frac{1}{2} \frac{f_i''}{(f_i')^4} \left( \frac{1}{1 - \frac{1}{2} u_i \frac{f_i''}{f_i'}} \right) \right] \epsilon_i^3. \quad (\text{A-13})$$

If the root is simple, the expression in outer brackets is bounded in some interval of  $\alpha$ . Thus, the order of the iteration formula (A-12) is three.

#### A.6 MODIFIED NEWTON-RAPHSON METHOD

Instead of considering the function  $f(x)$ , consider the function  $u(x) = f(x)/f'(x)$ . Both  $u(x)$  and  $f(x)$  have the same zeros. Applying the Newton-Raphson to find a zero of  $u(x)$ , we have the following iteration formula:

$$x_{i+1} = x_i - \frac{u_i}{u_i'} = x_i - u_i \frac{1}{\left(1 - \frac{u_i f_i'}{f_i}\right)}. \quad (\text{A-14})$$

This is also known as the modified Newton-Raphson method. This method is particularly useful in case of functions with multiple roots. The order of convergence of Newton-Raphson method is equal to one, when the function has a zero of multiplicity greater than 1. It may be noted that the second-order convergence of the Newton-Raphson method can still be maintained in such cases, by considering finding the zero for  $u(x)$  instead of  $f(x)$ , since  $u(x)$  has a zero of multiplicity 1, irrespective of the multiplicity of the zero of  $f(x)$ . The order of convergence for this method near a zero is the same as that for the Newton-Raphson method.

#### A.7 COMMENTS ON THE ORDER OF CONVERGENCE

We have considered four iterative methods. It is seen that the Newton-Raphson and modified Newton-Raphson methods exhibit second-order convergence near zero, whereas the other two methods, Method 2 and Halley's method, exhibit third-order convergence. The modified Newton-Raphson method retains the second-order convergence in case of a function with multiple roots. However, since this method uses the information on the second derivative of the function slightly improved convergence is likely to result when used for a function with simple roots.

## APPENDIX B

### COMPUTATION OF LOGARITHM

The logarithm function  $\log_{10}(x)$  is calculated in fixed-point arithmetic as follows:

- (1) Integer  $x$  is normalized and expressed as  
 $x = F \cdot 2^b$ , such that  $1/2 \leq F < 1$  and  $b \leq N-1$   
 $N$  = number of bits in the word used for representation  
of and integer (includes sign bit)
- (2) Calculate  $\log_{10}x$  as follows:

$$\log_{10}x = \log_{10}F + b \log_{10} 2$$

where  $\log_{10}F = (C_1z + C_3z^3 + C_5z^5) \log_{10} 2 - 1/2 \log_{10} 2^2$

$$C_1 = 2.8853913$$

$$C_3 = 0.96147063$$

$$C_5 = 0.59897865$$

$$\log_{10} 2^2 = 0.30102999$$

$$z = \frac{F - \sqrt{0.5}}{F + \sqrt{0.5}}$$

$$\sqrt{0.5} = 0.70710678$$

In the case of the subroutine LOG in the arithmetic package described in Appendix D, argument,  $X$ , is available as an integer in register  $R_0$ , and  $\log_{10}$  (argument) is available on return from the program in double precision in registers  $R_0$  and  $R_1$  with associated scale factor in register  $R_2$ .

A similar software program was written on a 24-bit wordlength machine with provisions to simulate varying wordlengths. Figure B-1

shows the plot of  $-\log_2 \epsilon$  versus the wordlength,  $N$ , where  $\epsilon$  is an average computed from the errors of the logarithm function computed for ten integers of varying magnitudes. Thus, the relationship  $\epsilon \approx 1/2^{N-5}$  can serve as a rough measure of the order of accuracy available from the logarithm function, which is computed as described in this appendix.

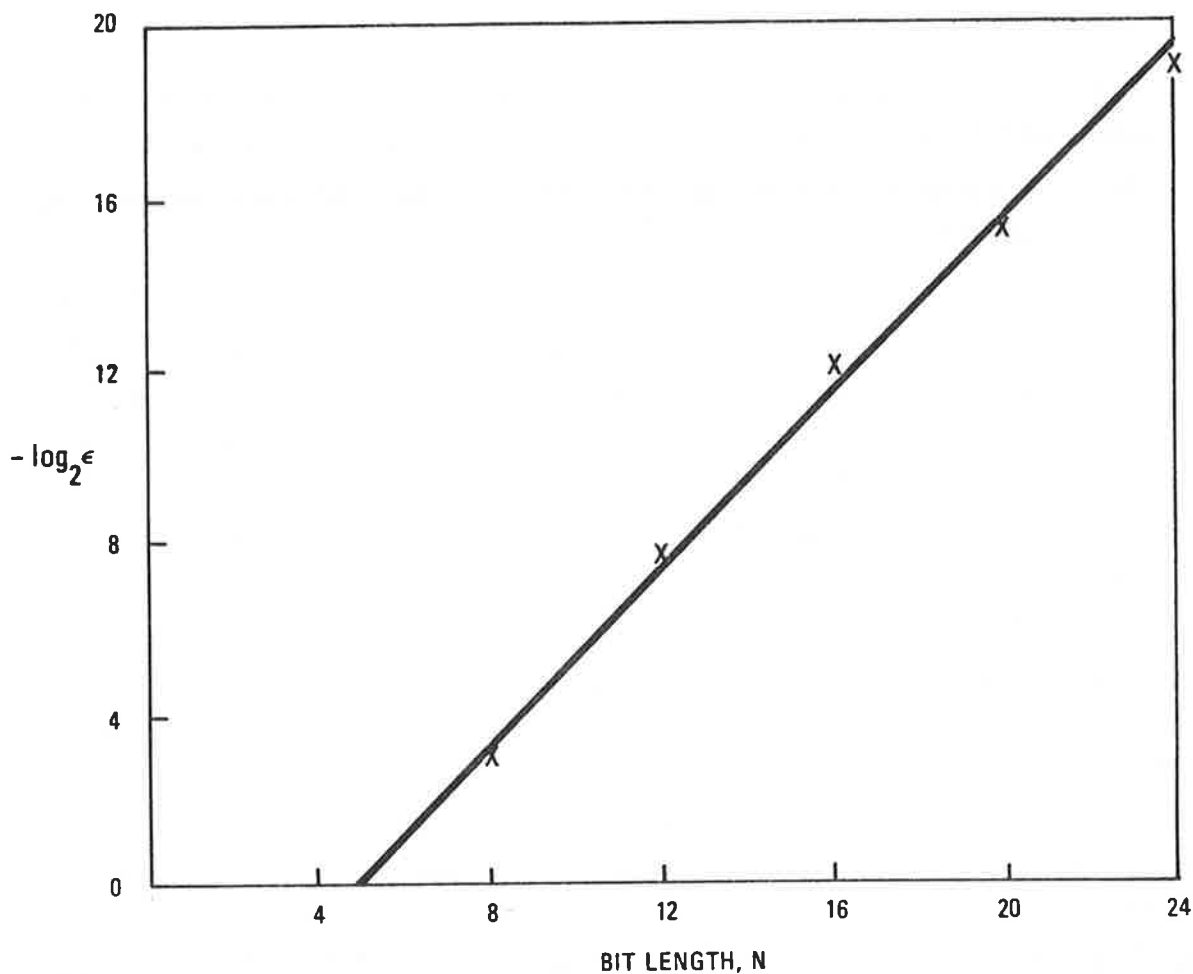


Figure B-1. Plot of  $-\log_2 \epsilon$  vs  $N$ , showing the dependence of  $\epsilon$ , the computation error for the fixed-point algorithm and  $N$ , bit length of the computer word.

## APPENDIX C

### ATMOSPHERIC TRANSMITTANCE FOR RVR RANGE 50 TO 5900 FEET

Table C-1 lists the atmospheric-transmittance values corresponding to a given set of RVR values. Table C-1 covers the following:

- (1) A range of RVR values from 50 feet to 5900 feet in varying increments.
- (2) Four transmissometer baselengths—500, 250, 75, and 50 feet.
- (3) Four discrete values of visual illuminance threshold\*  $s$ ,  $c$ ,  $E_t$ , namely, 2, 26, 260, and 2600 mile-candles, selected to broadly characterize the background luminance prevailing during night, dawn, twilight, normal day, and bright conditions, respectively.
- (4) Four values of runway-light intensity,  $I$  - 400, 2000, 10,000, and 20,000 candles.

The transmittance values for a given RVR and visual environment as characterized by  $s$ ,  $c$ , and  $E_t$  are calculated from Allard's law and Koschmieder's law, selecting the lower of the two transmittance values.

The transmittance values are positive and lie between 0 and 1. It may be noted, however, that some transmittance values are shown to be negative, especially for good daytime visibility conditions. The minus sign is included only to indicate that the RVR values correspond to the contrast visibility as projected by Koschmieder's law.

---

\*As recommended by ICAO All Weather Operational Panel (AWOP Report, IV-WP-213).

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 1 of 16)

BASE LINE 500.00000  
-----

RVR	ET = 2.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000000	.000000000
200.0	.000000000	.000000000	.000000000	.000000000
250.0	.000000000	.000000000	.000000000	.000000000
300.0	.000000000	.000000000	.000000001	.000000010
350.0	.000000001	.000000002	.000000022	.000000222
400.0	.000000016	.000000038	.000000281	.000002100
450.0	.000000151	.000000326	.000001951	.000011663
500.0	.000000897	.000001794	.000008968	.000044838
550.0	.000003782	.000007102	.000030675	.000132497
600.0	.000012375	.000022049	.000084307	.000322357
650.0	.000033382	.000056894	.000196217	.000676717
700.0	.000077488	.000127132	.000401347	.001267023
750.0	.000159675	.000253468	.000741145	.002167121
800.0	.000298930	.000461014	.001260579	.003446881
850.0	.000517446	.000777931	.002004934	.005167242
900.0	.000839478	.001233810	.003016929	.007377034
950.0	.001290092	.001858051	.004334475	.010111497
1100.0	.003651808	.005004259	.010400481	.021615588
1200.0	.006267910	.008366656	.016360231	.031990935
1300.0	.009847250	.012855700	.023874283	.044336862
1400.0	.014442376	.018499047	.032868623	.058400108
1500.0	.020058797	.025272501	.043215369	.073897241
1600.0	.026664214	.033113163	.054755631	.090543424
1700.0	.034198639	.041932143	.067317266	.0108070182
1800.0	.042583726	.051625361	.080727487	.0126234996
1900.0	.051730562	.062081989	.094821123	.0144825344
2000.0	.061545745	.073190638	.0109445532	.0163659243
2100.0	.071935856	.084843627	.0124463111	.0182583733
2200.0	.082810575	.096939708	.0139752159	.0201472301
2300.0	.094084719	.109385661	.0155206676	.0220221846
2900.0	.166191306	.187288032	.0247184990	.0326237714
3400.0	.226747572	.251079793	.0318127688	.0403079931
3900.0	.284082032	.310482263	.0381635612	.0469094599
4400.0	.336867263	.364474175	.0437617994	.0525440544
4900.0	.384783789	.412984843	.0486695783	.0573562901
5400.0	.427993336	.456362714	.0529699545	.0614821497
5900.0	.466863054	.495108661	.0567460020	.0650384250

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 2 of 16)

BASE LINE 500.00000  
-----

RVK	I = 20000.0	ET = 26.00000 I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000000	.000000000
200.0	.000000000	.000000000	.000000000	.000000000
250.0	.000000000	.000000000	.000000001	.000000021
300.0	.000000001	.000000003	.000000051	.000000074
350.0	.000000032	.000000087	.000000086	.000000864
400.0	.000000390	.000000927	.000000693	.000005184
450.0	.000002611	.000005640	.000033720	.000201615
500.0	.000011658	.000023316	.000116578	.000582889
550.0	.000038937	.000073119	.000315831	.001364214
600.0	.000104910	.000186928	.000714740	.002732898
650.0	.000240097	.000409212	.001411293	.004867280
700.0	.000484070	.000794200	.002507229	.007915134
750.0	.000882806	.001401367	.004097623	.011981523
800.0	.001485199	.002290490	.006263033	.017125408
850.0	.002339513	.003517240	.009064854	.023362517
900.0	.003490334	.005129864	.012543609	.030671794
950.0	.004976302	.007167101	.016719469	.039003307
1100.0	.011717794	.016057490	.033372697	.069359339
1200.0	.018250125	.024360994	.047635694	.093147242
1300.0	.026409167	.034477476	.064028021	.118906254
1400.0	.036097370	.046236641	.082152053	.145965616
1500.0	.047164946	.059424108	.101613796	.173757147
1600.0	.059434154	.073808768	.122049523	.201820004
1700.0	.072717547	.089161518	.143138634	.229792728
1800.0	.086830513	.105266894	.164607699	.257399962
1900.0	.101599176	.121929448	.186229330	.284437961
2000.0	.116864842	.138976501	.207818342	.310760904
2100.0	.132486011	.156258565	.229226731	.336268890
2200.0	.148338794	.173648346	.250338399	.360897845
2300.0	.164316317	.191038982	.271064095	.384611262
2900.0	.258623212	.291453466	.384663779	.507683868
3400.0	.330641867	.366122957	.463891770	.587768591
3900.0	.394690811	.431370662	.530227372	.651738959
4400.0	.450861633	.487810600	.585705960	.703247267
4900.0	.499901518	.536539626	.632303043	.743816110
5400.0	.542725103	.578699434	.671695599	.764481386
5900.0	.580218385	.615322084	.705240506	.782079812



TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 3 of 16)

BASE LINE 500.00000  
-----

RVR	ET = 260.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000000	.000000000
200.0	.000000000	.000000000	.000000000	.000000027
250.0	.000000001	.000000003	.000000085	.000002123
300.0	.000000051	.000000161	.000002353	.000034394
350.0	.000000868	.000002335	.000023274	.000231954
400.0	.000006934	.000016492	.000123309	.000921949
450.0	.000033720	.000072840	.000435512	.002603958
500.0	.000116578	.000233155	.001165777	.005828885
550.0	.000315831	.000593088	.002561806	.011065564
600.0	.000714740	.001273523	.004869470	.018619014
650.0	.001411293	.002406352	.008295601	.028609950
700.0	.002507229	.004113537	.012986129	.040996237
750.0	.004097623	.006504571	.019019482	.055613303
800.0	.006263033	.009658917	.026410990	.072217247
850.0	.009064854	.013628161	.035123363	.090522162
900.0	.012543609	.018435775	.045079394	.110228714
950.0	.016719469	.024080157	.056174378	.131044024
1100.0	.033372697	.045732308	.095046639	.197537886
1200.0	.047635694	.063586023	.124336651	.243128946
1300.0	.064028021	.083589331	.155233506	.288283697
1400.0	.082152053	.105227471	.186965415	.332195253
1500.0	.101613796	.128025360	.218920287	.374348425
1600.0	.122049523	.151568153	.250631749	-.403983233
1700.0	.143138634	.175507266	.281756870	-.426106774
1800.0	.164607699	.199558204	.312052683	-.446787225
1900.0	.186229330	.223494326	.341354770	-.4666140135
2000.0	.207818342	.247139050	.369559078	-.484273464
2100.0	.229226731	.270357903	.396607113	-.501286899
2200.0	.250338399	.293051115	.422474205	-.517271788
2300.0	.271064095	.315147088	.447160363	-.532311443
2900.0	.384663779	.433493926	.572130482	-.606486521
3400.0	.463891770	.513671872	.650841881	-.652768545
3900.0	.530227372	.579503059	-.689458352	-.689458352
4400.0	.585705960	.633705676	-.719216093	-.719216093
4900.0	.632303043	.678644945	-.743816110	-.743816110
5400.0	.671695599	.716218691	-.764481386	-.764481386
5900.0	.705240506	.747908149	-.782079812	-.782079812

TABLE C-1 ATMOSPHERIC TRANSMITTANCE (page 4 of 16)

BASE LINE 500.00000  
-----

RVR	ET = 2600.00000 I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000000	.000000025
200.0	.000000000	.000000003	.000000150	.000008400
250.0	.000000085	.000000340	.000008494	.000212349
300.0	.000002353	.000007469	.000109193	.001596418
350.0	.000023274	.000062649	.000624374	.006222626
400.0	.000123309	.000293280	.002192776	.016394828
450.0	.000435512	.000940759	.005624860	.033631408
500.0	.001165777	.002331554	.011657771	-.055000000
550.0	.002561806	.004810718	.020779599	-.071593760
600.0	.004869470	.008676409	.033175311	-.089187141
650.0	.008295601	.014138697	.048761677	-.107410867
700.0	.012986129	.021305961	.067261325	-.125967661
750.0	.019019482	.030191546	.088280616	-.144624474
800.0	.026410990	.040731314	.111374220	-.163202453
850.0	.035123363	.052804698	.136091621	-.181566983
900.0	.045079394	.066254740	.162006942	-.199618825
950.0	.056174378	.080904953	.188735703	-.217286625
1100.0	.095046639	.130247255	-.267570103	-.267570103
1200.0	.124336651	.165969517	-.298642162	-.298642162
1300.0	.155233506	.202659158	-.327735972	-.327735972
1400.0	.186965415	.239481512	-.354919231	-.354919231
1500.0	.218920287	.275822278	-.380295246	-.380295246
1600.0	.250631749	.311248996	-.403983233	-.403983233
1700.0	.281756870	.345471916	-.426106774	-.426106774
1800.0	.312052683	.378309602	-.446787225	-.446787225
1900.0	.341354770	.409660788	-.466140135	-.466140135
2000.0	.369559078	.439482285	-.484273464	-.484273464
2100.0	.396607113	.467772092	-.501286899	-.501286899
2200.0	.422474205	.494556718	-.517271788	-.517271788
2300.0	.447160363	.519881787	-.532311443	-.532311443
2900.0	.572130482	-.606486521	-.606486521	-.606486521
3400.0	.650841881	-.652768545	-.652768545	-.652768545
3900.0	-.689458352	-.689458352	-.689458352	-.689458352
4400.0	-.719216093	-.719216093	-.719216093	-.719216093
4900.0	-.743816110	-.743816110	-.743816110	-.743816110
5400.0	-.764481386	-.764481386	-.764481386	-.764481386
5900.0	-.782079812	-.782079812	-.782079812	-.782079812

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 5 of 16)

BASE LINE 250.00000  
 -----

RVR	I= 20000.0	ET= 2.00000 I= 10000.0	I= 2000.0	I= 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000000	.000000001
200.0	.000000003	.000000007	.000000050	.0000000371
250.0	.000000224	.000000448	.000002242	.000011209
300.0	.000003898	.000006945	.000026555	.000101536
350.0	.000028787	.000047229	.000149100	.000470697
400.0	.000125685	.000193832	.000530008	.001449235
450.0	.000388626	.000571177	.001396649	.003415104
500.0	.000946970	.001339217	.002994581	.006696087
550.0	.001944663	.002664871	.005538470	.011510745
600.0	.003517746	.004695627	.009181869	.017954305
650.0	.005777679	.007542828	.014007763	.026013777
700.0	.008802727	.011275296	.020033650	.035595265
750.0	.012636251	.015920678	.027223976	.046552345
800.0	.017289604	.021471229	.035504634	.058710147
850.0	.022747431	.027891418	.044776485	.071883533
900.0	.028973754	.035125637	.054926579	.085889660
950.0	.035917852	.043105112	.065836730	.100555940
1100.0	.060430193	.070740788	.101982748	.147022407
1200.0	.079170135	.091469429	.127907119	.178860100
1300.0	.099233309	.113382977	.154513050	.210563201
1400.0	.120176438	.136011203	.181297057	.241661144
1500.0	.141629084	.158973271	.207883065	.271840470
1600.0	.163291807	.181970224	.233999211	.300904344
1700.0	.184928740	.204773394	.259455711	.328740296
1800.0	.206358246	.227212149	.284125830	.355295645
1900.0	.227443537	.249162574	.307930386	.380559252
2000.0	.248084150	.270537684	.330825531	.404548196
2100.0	.268208605	.291279293	.352793298	.427298179
2200.0	.287768266	.311351422	.373834401	.448856659
2300.0	.306732325	.330735032	.393962785	.469278005
2900.0	.407665680	.432767873	.497177021	.571172228
3400.0	.476180189	.501078630	.564028091	.634885762
3900.0	.532993463	.557209712	.617766632	.684904810
4400.0	.580402673	.603716966	.661527017	.724872778
4900.0	.620309430	.642638968	.697635853	.757339357
5400.0	.654211996	.675546234	.727804606	.784105540
5900.0	.683273777	.703639581	.753299423	.806464041

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 6 of 16)

BASE LINE 250.00000  
 -----

RVR	I = 20000.0	ET = 26.00000 I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000000
150.0	.000000000	.000000000	.000000005	.000000074
200.0	.000000069	.000000164	.000001226	.0000009165
250.0	.000002914	.000005829	.000029144	.000145722
300.0	.000033044	.000058878	.000225129	.000860809
350.0	.000179831	.000295044	.000931432	.002940461
400.0	.000624449	.000963032	.002633281	.007200347
450.0	.001615806	.002374806	.005806905	.014199119
500.0	.003414348	.004828617	.010797116	.024143085
550.0	.006239966	.008550944	.017771647	.036935273
600.0	.010242536	.013672145	.026734630	.052277122
650.0	.015495056	.020228977	.037567176	.069765894
700.0	.022001594	.028181548	.050072237	.088967040
750.0	.029712054	.037434842	.064012680	.109460143
800.0	.038538281	.047859065	.079139324	.130864082
850.0	.048368515	.059306321	.095209525	.152848018
900.0	.059079047	.071623068	.111998255	.175133647
950.0	.070542906	.084658733	.129303786	.197492550
1100.0	.108248759	.126718153	.182681956	.263361613
1200.0	.135093022	.156080087	.218256029	.305200331
1300.0	.162508974	.185681114	.253037589	.344827862
1400.0	.189993079	.215027071	.286621794	.382054468
1500.0	.217174920	.243770606	.318769189	.416841873
1600.0	.243791210	.271677692	.349355868	.449243814
1700.0	.269661912	.298599260	.378336667	.479367007
1800.0	.294670176	.324448600	.405718744	.507345998
1900.0	.318746257	.349183974	.431542965	.533327255
2000.0	.341855001	.372795522	.455870970	.557459329
2100.0	.363986278	.395295541	.478776285	.579886963
2200.0	.385147756	.416711346	.500338284	.600747739
2300.0	.405359491	.437080063	.520638162	.620170349
2900.0	.508550107	.539864303	.620212688	.712519381
3400.0	.575014667	.605080951	.681096007	.766660675
3900.0	.628244229	.656788141	.728167132	.807303511
4400.0	.671462309	.698434392	.765314288	.838598394
4900.0	.707037141	.732488653	.795174851	.862447744
5400.0	.736698787	.760722968	.819570374	.874346262
5900.0	.761720674	.784424683	.839785988	.884352764

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 7 of 16)

BASE LINE 250.00000  
-----

RVR	ET = 260.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000000	.000000001
150.0	.000000005	.000000016	.000000233	.000003412
200.0	.000001226	.000002915	.000021798	.000162979
250.0	.000029144	.000058289	.000291444	.001457221
300.0	.000225129	.000401134	.001533787	.005864622
350.0	.000931432	.001528173	.004824328	.015230043
400.0	.002633281	.004061074	.011104453	.030363612
450.0	.005806905	.008534609	.020868936	.051028989
500.0	.010797116	.015269428	.034143478	.076347138
550.0	.017771647	.024353394	.050614288	.105192982
600.0	.026734630	.035686449	.069781586	.136451508
650.0	.037567176	.049044387	.091080191	.169144760
700.0	.050072237	.064136862	.113956697	.202475275
750.0	.064012680	.080650923	.137911139	.235824730
800.0	.079139324	.098279788	.162514583	.268732668
850.0	.095209525	.116739716	.187412281	.300869012
900.0	.111998255	.135778404	.212319086	.332007099
950.0	.129303786	.155177825	.237011346	.362000034
1100.0	.182681956	.213851136	.308296349	.444452343
1200.0	.218256029	.252162692	.352614026	.493081075
1300.0	.253037589	.289118195	.393996835	.536920569
1400.0	.286621794	.324387841	.432394976	.576363820
1500.0	.318769189	.357806317	.467889182	.611840195
1600.0	.349355868	.389317548	.500631351	.635596753
1700.0	.378336667	.418935874	.530807753	.652768545
1800.0	.405718744	.446719380	.558616759	.668421443
1900.0	.431542965	.472751865	.584255740	.682744560
2000.0	.455870970	.497130818	.607913709	.695897596
2100.0	.478776285	.519959521	.629767507	.708016171
2200.0	.500338284	.541341958	.649980157	.719216093
2300.0	.520638162	.561379629	.668700503	.729596767
2900.0	.620212688	.658402556	.756393074	.778772445
3400.0	.681096007	.716709057	.806747718	.807940929
3900.0	.728167132	.761250983	.830336289	.830336289
4400.0	.765314288	.796056327	.848066090	.848066090
4900.0	.795174851	.823799093	.862447744	.862447744
5400.0	.819570374	.846297046	.874346262	.874346262
5900.0	.839785988	.864816829	.884352764	.884352764

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 8 of 16)

BASE LINE 250.00000  
-----

RVR	ET= 2600.00000			
	I= 20000.0	I= 10000.0	I= 2000.0	I= 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000000000	.000000000	.000000005	.000000262
150.0	.000000233	.000000741	.000010833	.000158385
200.0	.000021798	.000051845	.000387632	.002898224
250.0	.000291444	.000582889	.002914443	.014572214
300.0	.001533787	.002732898	.010449568	.039955204
350.0	.004824328	.007915134	.024987486	.078883625
400.0	.011104453	.017125408	.046827091	.128042290
450.0	.020868936	.030671794	.074999066	.183388679
500.0	.034143478	.048286170	.107971158	.234520788
550.0	.050614288	.069359339	.144151307	.267570103
600.0	.069781586	.093147242	.182140911	.298642162
650.0	.091080191	.118906254	.220820463	.327735972
700.0	.113956697	.145965616	.259347884	.354919231
750.0	.137911139	.173757147	.297120542	.380295246
800.0	.162514583	.201820004	.333727763	.403983233
850.0	.187412281	.229792728	.368905979	.426106774
900.0	.212319086	.257399962	.402500860	.446787225
950.0	.237011346	.284437961	.434437225	.466140135
1100.0	.308296349	.360897845	.517271788	.517271788
1200.0	.352614026	.407393566	.546481621	.546481621
1300.0	.393996835	.450176807	.572482289	.572482289
1400.0	.432394976	.489368482	.595750981	.595750981
1500.0	.467889182	.525187850	.616680830	.616680830
1600.0	.500631351	.557896940	.635596753	.635596753
1700.0	.530807753	.587768591	.652768545	.652768545
1800.0	.558616759	.615068779	.668421443	.668421443
1900.0	.584255740	.640047489	.682744560	.682744560
2000.0	.607913709	.662934601	.695897596	.695897596
2100.0	.629767507	.683938661	.708016171	.708016171
2200.0	.649980157	.703247267	.719216093	.719216093
2300.0	.668700503	.721028285	.729596767	.729596767
2900.0	.756393074	.778772445	.778772445	.778772445
3400.0	.806747718	.807940929	.807940929	.807940929
3900.0	.830336289	.830336289	.830336289	.830336289
4400.0	.848066090	.848066090	.848066090	.848066090
4900.0	.862447744	.862447744	.862447744	.862447744
5400.0	.874346262	.874346262	.874346262	.874346262
5900.0	.884352764	.884352764	.884352764	.884352764

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 9 of 16)

BASE LINE 75.00000  
-----

RVR	I = 20000.0	ET = 2.00000 I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000000	.000000000
100.0	.000002606	.000004384	.000014657	.000049009
150.0	.000284091	.000401765	.000898374	.002008826
200.0	.002715170	.003521140	.006438683	.011773641
250.0	.010120073	.012459272	.020192201	.032724623
300.0	.023836565	.028346612	.042388072	.063384952
350.0	.043426765	.050380666	.071128425	.100420524
400.0	.067574801	.076953415	.104060229	.140715407
450.0	.094811175	.106421946	.139163774	.181978968
500.0	.123851373	.137421702	.174944714	.222713388
550.0	.153692721	.168928469	.210386048	.262017940
600.0	.183602813	.200220288	.244838287	.299399163
650.0	.213071824	.230812993	.277914137	.334627034
700.0	.241760401	.260398442	.309405497	.367635692
750.0	.269454517	.288794201	.339223139	.398457925
800.0	.296029855	.315905418	.367354625	.427182989
850.0	.321425040	.341697008	.393835627	.453929936
900.0	.345622041	.366173798	.418730663	.478831006
950.0	.368632064	.389366545	.442120321	.502021502
1100.0	.430899703	.451752919	.504147959	.562619861
1200.0	.467270556	.487958391	.539595620	.596697256
1300.0	.500031359	.520432437	.571070268	.626635135
1400.0	.529597343	.549632500	.599124703	.653073480
1500.0	.556346978	.575966512	.624231577	.676541176
1600.0	.580616426	.599791213	.646791521	.697474826
1700.0	.602700183	.621415467	.667143057	.716235565
1800.0	.622854417	.641105444	.685572226	.733123204
1900.0	.641301240	.659090183	.702321177	.748387776
2000.0	.658233082	.675566850	.717595498	.762238851
2100.0	.673816765	.690705451	.731570354	.774852989
2200.0	.688197150	.704652920	.744395568	.786379714
2300.0	.701500305	.717536602	.756199802	.796946302
2900.0	.763996282	.777815340	.810873886	.845337480
3400.0	.800444817	.812777672	.842151488	.872586875
3900.0	.827973744	.839084316	.865460665	.892666146
4400.0	.849412839	.859508180	.883413996	.907984714
4900.0	.866527829	.875770107	.897611974	.919998582
5400.0	.880472003	.888989278	.909084891	.929634766
5900.0	.892027875	.899922434	.918523539	.937509122

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 10 of 16)

BASE LINE 75.00000  
-----

RVR	I = 20000.0	ET = 26.00000 I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000000	.000000001	.000000014
100.0	.000017845	.000030011	.000100348	.000335532
150.0	.001024304	.001448585	.003239135	.007242925
200.0	.007104374	.009213233	.016847126	.030806302
250.0	.021845735	.026895254	.043587971	.070641133
300.0	.045261559	.053825367	.080487698	.120357180
350.0	.075241878	.087290314	.123238199	.173990136
400.0	.109307297	.124477907	.168325206	.227617700
450.0	.145384048	.163188076	.213394599	.279047685
500.0	.181966871	.201904885	.257035037	.327218484
550.0	.218049285	.239664777	.298482108	.371734094
600.0	.253001008	.275899555	.337382266	.412566064
650.0	.286456017	.310307433	.373630709	.449876128
700.0	.318226426	.342759464	.407266885	.483914619
750.0	.348240922	.373235378	.438409346	.514963922
800.0	.376502359	.401780878	.467215994	.543308049
850.0	.403059188	.428479744	.493860305	.569217107
900.0	.427986474	.453435874	.518517451	.592940175
950.0	.451373444	.476761887	.541356521	.614702833
1100.0	.513247542	.538085948	.600494033	.670140310
1200.0	.548516726	.572801636	.633417233	.700447356
1300.0	.579779966	.603434755	.662148672	.726575425
1400.0	.607604994	.630591252	.687373465	.749268690
1500.0	.632474327	.654778485	.709647866	.769115213
1600.0	.654795088	.676419616	.729424612	.786583137
1700.0	.674910031	.695867604	.747073842	.802048151
1800.0	.693107909	.713417521	.762899836	.815814221
1900.0	.709632550	.729316923	.777154225	.828129267
2000.0	.724690522	.743774365	.790046369	.839197065
2100.0	.738457496	.756966351	.801751515	.849186348
2200.0	.751083483	.769042954	.812417221	.858237810
2300.0	.762697115	.780132371	.822168435	.866469539
2900.0	.816394603	.831161434	.866487285	.903314548
3400.0	.847039526	.860090289	.891173985	.923381047
3900.0	.869838352	.881510705	.909220714	.937801779
4400.0	.887373575	.897920083	.922894263	.948563060
4900.0	.901223833	.910836174	.933552596	.956576786
5400.0	.912403592	.921229759	.942054168	.960516959
5900.0	.921592057	.929748262	.948965857	.963801641



TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 11 of 16)

BASE LINE 75.00000  
-----

RVR	ET= 260.00000			
	I= 20000.0	I= 10000.0	I= 2000.0	I= 400.0
50.0	.000000001	.000000004	.000000040	.000000445
100.0	.000100348	.000168764	.000564296	.001886836
150.0	.003239135	.004580828	.010243043	.022904141
200.0	.016847126	.021848019	.039950831	.073053254
250.0	.043587971	.053663087	.086969437	.140947591
300.0	.080487698	.095716543	.143129615	.214028696
350.0	.123238199	.142972256	.201851070	.284977349
400.0	.168325206	.191686831	.259208450	.350514534
450.0	.213394599	.239527338	.313220436	.409585987
500.0	.257035037	.285198231	.363071640	.462208394
550.0	.298482108	.328071004	.408584549	.508857325
600.0	.337382266	.367917970	.449906483	.550165688
650.0	.373630709	.404740621	.487334524	.586783056
700.0	.407266885	.438664321	.521221061	.619315000
750.0	.438409346	.469875502	.551924666	.648301168
800.0	.467215994	.498585063	.579785969	.674211474
850.0	.493860305	.525007600	.605117084	.697450256
900.0	.518517451	.549350103	.628198233	.718363422
950.0	.541356521	.571806251	.649278080	.737246269
1100.0	.600494033	.629554697	.702571477	.784056862
1200.0	.633417233	.661461010	.731458810	.808863988
1300.0	.662148672	.689164070	.756219409	.829799201
1400.0	.687373465	.713377438	.777614216	.847635258
1500.0	.709647866	.734673544	.796238002	.862961462
1600.0	.729424612	.753513771	.812559950	.872879916
1700.0	.747073842	.770272273	.826953667	.879888751
1800.0	.762899836	.785254508	.839719403	.886166059
1900.0	.777154225	.798711569	.851100599	.891820548
2000.0	.790046369	.810851279	.861296299	.896940431
2100.0	.801751515	.821846785	.870470535	.901598028
2200.0	.812417221	.831843269	.878759493	.905853192
2300.0	.822168435	.840963206	.886277033	.909755878
2900.0	.866487285	.882160185	.919653574	.927733359
3400.0	.891173985	.904904750	.937608042	.938023854
3900.0	.909220714	.921421539	.945749686	.945749686
4400.0	.922894263	.933862937	.951763202	.951763202
4900.0	.933552596	.943509751	.956576786	.956576786
5400.0	.942054168	.951167160	.960516959	.960516959
5900.0	.948965857	.957364324	.963801641	.963801641

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 12 of 16)

BASE LINE 75.00000  
-----

RVR	ET= 1000.00000			
	I= 20000.0	I= 10000.0	I= 2000.0	I= 400.0
50.0	.000000009	.000000027	.000000300	.000000357
100.0	.000275598	.000463499	.001549803	.005182080
150.0	.006352466	.008983743	.020088261	.044918717
200.0	.027919704	.036207376	.066208051	.121066659
250.0	.065294207	.080386598	.130279071	.211137636
300.0	.112716155	.134042854	.200440818	.299728933
350.0	.164478144	.190815928	.269397717	.380341046
400.0	.216691388	.246765690	.333688816	.451230583
450.0	.267108596	.299819262	.392061801	.512683726
500.0	.314591023	.349060595	.444371631	.565707356
550.0	.358669679	.394225043	.490973781	.611466111
600.0	.399255201	.435390884	.532415368	.651061228
650.0	.436461197	.472802614	.569285673	.685457669
700.0	.470500973	.506773316	.602148187	.715472632
750.0	.501628807	.537632443	.631513252	.741787429
800.0	.530107502	.565699132	.657830416	-.761919348
850.0	.556191207	.591269652	.681489883	-.774204215
900.0	.580117008	.614612560	.702827800	-.785290284
950.0	.602101469	.635967926	.722132773	-.795343890
1100.0	.658259108	.690115290	.770155986	-.820570005
1200.0	.689055242	.719562323	.795708579	-.834204769
1300.0	.715660504	.744859163	.817333609	-.845918683
1400.0	.738811210	.766761121	.835804885	-.856090011
1500.0	.759091684	.785860995	.851714878	-.865004037
1600.0	.776968724	.802628022	.865522848	-.872879916
1700.0	.792818051	.817436949	.877589011	-.879888751
1800.0	.806944437	.830589714	-.886166059	-.886166059
1900.0	.819597018	.842331676	-.891820548	-.891820548
2000.0	.830980975	.852863848	-.896940431	-.896940431
2100.0	.841266506	.862352188	-.901598028	-.901598028
2200.0	.850595755	.870934706	-.905853192	-.905853192
2300.0	.859088203	.878726960	-.909755878	-.909755878
2900.0	.897206050	.913434587	-.927733359	-.927733359
3400.0	.918052462	.932197357	-.938023854	-.938023854
3900.0	.933082048	.945603068	-.945749686	-.945749686
4400.0	.944330445	-.951763202	-.951763202	-.951763202
4900.0	.953000833	-.956576786	-.956576786	-.956576786
5400.0	.959845306	-.960516959	-.960516959	-.960516959
5900.0	-.963801641	-.963801641	-.963801641	-.963801641

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 13 of 16)

BASE LINE 60.00000  
-----

RVR	ET = 26.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000005	.000000011	.000000076	.000000523
100.0	.000158930	.000240893	.000632711	.001661833
150.0	.004058291	.005354947	.010193967	.019405785
200.0	.019108238	.023525001	.038125948	.061789069
250.0	.046934622	.055429444	.081563025	.120017929
300.0	.084058881	.096558298	.133224348	.183813584
350.0	.126231488	.142158504	.187325230	.246842367
400.0	.170184234	.188831230	.240391620	.306030579
450.0	.213802808	.234504260	.290634213	.360199196
500.0	.255854973	.278046518	.337282542	.409138419
550.0	.295694831	.318921185	.380132185	.453091500
600.0	.333041076	.356944587	.419273874	.492487034
650.0	.367829332	.392133317	.454939924	.527806044
700.0	.400118833	.424611316	.487419925	.559519199
750.0	.430035039	.454554794	.517015833	.588059734
800.0	.457735042	.482160258	.544019117	.613814172
850.0	.483387105	.507626443	.568700089	.637121638
900.0	.507158881	.531144516	.591303843	.658277031
950.0	.529210928	.552893169	.612049712	.677535682
1100.0	.586491203	.609089791	.664977509	.725993268
1200.0	.618517369	.640329337	.693987902	.752142969
1300.0	.646562576	.667581443	.719058807	.774505603
1400.0	.671269736	.691509822	.740890815	.793798123
1500.0	.693161321	.712648739	.760036275	.810574844
1600.0	.712663194	.731430313	.776934363	.825269330
1700.0	.730124138	.748206186	.791937347	.838224506
1800.0	.745831435	.763264418	.805330259	.849714477
1900.0	.760023153	.776842599	.817345677	.859960506
2000.0	.772897789	.789138015	.828174902	.869142857
2100.0	.784621858	.800315558	.837976484	.877409642
2200.0	.795335869	.810513920	.846882769	.884883536
2300.0	.805159058	.819850450	.855004970	.891666888
2900.0	.850198041	.862478558	.891681498	.921873230
3400.0	.875634655	.886411201	.911947842	.938220169
3900.0	.894439236	.904028397	.926692093	.949923960
4400.0	.908835314	.917466357	.937824463	.958634305
4900.0	.920165900	.928009060	.946479109	.965107898
5400.0	.929286417	.936471073	.953368285	.968286841
5900.0	.936765695	.943392241	.958959950	.970934938

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 14 of 16)

BASE LINE 60.00000  
-----

RVR	ET = 260.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000076	.000000174	.000001202	.000008293
100.0	.000632711	.000959010	.002518867	.006615875
150.0	.010193967	.013451020	.025606087	.048745128
200.0	.038125948	.046938548	.076071267	.123285401
250.0	.081563025	.096325334	.141740293	.208567256
300.0	.133224348	.153034589	.211146362	.291324899
350.0	.187325230	.210960631	.277987229	.366309578
400.0	.240391620	.266731202	.339562188	.432279683
450.0	.290634213	.318774865	.395075475	.489639078
500.0	.337282542	.366536697	.444624983	.539349478
550.0	.380132185	.409990957	.488681109	.582474375
600.0	.419273874	.449366611	.527834534	.620004442
650.0	.454939924	.484999661	.562680343	.652802865
700.0	.487419925	.517256373	.593769061	.681599525
750.0	.517015833	.546495062	.621589748	.707003303
800.0	.544019117	.573048540	.646567932	.729519511
850.0	.568700089	.597217427	.669069960	.749567228
900.0	.591303843	.619269040	.689409666	.767494671
950.0	.612049712	.639438997	.707855470	.783592132
1100.0	.664977509	.690600319	.753967128	.823148229
1200.0	.693987902	.718461333	.778667234	.843918292
1300.0	.719058807	.742434427	.799683722	.861347525
1400.0	.740890815	.763230111	.817732678	.876127295
1500.0	.760036275	.781403804	.833363204	.888777642
1600.0	.776934363	.797393985	.847001794	.896940431
1700.0	.791937347	.811550243	.858983738	.902697444
1800.0	.805330259	.824153961	.869575611	.907845802
1900.0	.817345677	.835433681	.878991585	.912477112
2000.0	.828174902	.845576617	.887405396	.916665488
2100.0	.837976484	.854737363	.894959248	.920471535
2200.0	.846882769	.863044532	.901770499	.923945288
2300.0	.855004970	.870605880	.907936752	.927128425
2900.0	.891681498	.904561214	.935189044	.941756309
3400.0	.911947842	.923171299	.949766963	.950103912
3900.0	.926692093	.936627032	.956359033	.956359033
4400.0	.937824463	.946730812	.961220728	.961220728
4900.0	.946479109	.954546554	.965107898	.965107898
5400.0	.953368285	.960739127	.968286841	.968286841
5900.0	.958959950	.965743496	.970934938	.970934938

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 15 of 16)

BASE LINE 60.00000  
-----

RVR	ET = 2600.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000001202	.000002762	.000019052	.000131432
100.0	.002518867	.003817889	.010027792	.026338274
150.0	.025606087	.033787434	.064319582	.122442225
200.0	.076071267	.093654715	.151782132	.245986714
250.0	.141740293	.167394246	.246316398	.362448350
300.0	.211146362	.242543479	.334644432	.461718849
350.0	.277987229	.313061735	.412527986	.543596743
400.0	.339562188	.376767838	.479644339	.610611282
450.0	.395075475	.433328650	.537048372	.665594010
500.0	.444624983	.483189470	.586129880	-.706063112
550.0	.488681109	.527066225	.628226801	-.728760828
600.0	.527834534	.565719046	.664504308	-.748231985
650.0	.562680343	.599858929	.695936213	-.765113362
700.0	.593769061	.630115461	.723322293	-.779885934
750.0	.621589748	.657031577	.747315247	-.792919348
800.0	.646567932	.681069467	.768447428	-.804502136
850.0	.669069960	.702620323	.787154108	-.814862675
900.0	.689409666	.722014692	.803792658	-.824184000
950.0	.707855470	.739532072	.818657957	-.832614480
1100.0	.753967128	.783018872	-.853674896	-.853674896
1200.0	.778667234	.806126874	-.865004037	-.865004037
1300.0	.799683722	.825680348	-.874707587	-.874707587
1400.0	.817732678	.842388906	-.883111507	-.883111507
1500.0	.833363204	.856792233	-.890460189	-.890460189
1600.0	.847001794	.869306556	-.896940431	-.896940431
1700.0	.858983738	.880257086	-.902697444	-.902697444
1800.0	.869575611	.889900977	-.907845802	-.907845802
1900.0	.878991585	.898443824	-.912477112	-.912477112
2000.0	.887405396	.906051669	-.916665488	-.916665488
2100.0	.894959248	.912859873	-.920471535	-.920471535
2200.0	.901770499	.918979731	-.923945288	-.923945288
2300.0	.907936752	.924503485	-.927128425	-.927128425
2900.0	.935189044	-.941756309	-.941756309	-.941756309
3400.0	.949766963	-.950103912	-.950103912	-.950103912
3900.0	-.956359033	-.956359033	-.956359033	-.956359033
4400.0	-.961220728	-.961220728	-.961220728	-.961220728
4900.0	-.965107898	-.965107898	-.965107898	-.965107898
5400.0	-.968286841	-.968286841	-.968286841	-.968286841
5900.0	-.970934938	-.970934938	-.970934938	-.970934938

TABLE C-1. ATMOSPHERIC TRANSMITTANCE (page 16 of 16)

BASE LINE 60.00000

RVR	ET = 2.00000			
	I = 20000.0	I = 10000.0	I = 2000.0	I = 400.0
50.0	.000000000	.000000001	.000000003	.000000024
100.0	.000034107	.000051696	.000135781	.000356633
150.0	.001454675	.001919455	.003653978	.006955909
200.0	.008851923	.010897996	.017661909	.028623889
250.0	.025359864	.029949813	.044070393	.064848468
300.0	.050326292	.057809729	.079761798	.110049718
350.0	.081321613	.091582212	.120679793	.159022283
400.0	.115831804	.128523434	.163616772	.208292351
450.0	.151875466	.166580805	.206452886	.255868580
500.0	.188070492	.204382760	.247925193	.300744061
550.0	.223523217	.241080607	.287351553	.342503349
600.0	.257693501	.276189056	.324416897	.381066232
650.0	.290282476	.309462623	.359028158	.416532429
700.0	.321150154	.340808726	.391221235	.449090776
750.0	.350258251	.370229290	.421103038	.478967422
800.0	.377631670	.397782488	.448816081	.506397042
850.0	.403333008	.423558051	.474517245	.531607451
900.0	.427445889	.447661568	.498365318	.554811957
950.0	.450064218	.470204636	.520513959	.576206104
1100.0	.509918720	.529566829	.578157828	.631207350
1200.0	.544069941	.563256526	.610456514	.661611785
1300.0	.574377995	.593050239	.638780485	.688037000
1400.0	.601390604	.619523698	.663764133	.711163794
1500.0	.625570970	.643158164	.685924929	.731535467
1600.0	.647308716	.664354803	.705685923	.749588353
1700.0	.666931237	.683448267	.723394457	.765675421
1800.0	.684714049	.700718481	.739337223	.780084363
1900.0	.700889678	.716400490	.753752232	.793051422
2000.0	.715655090	.730692525	.766838245	.804772012
2100.0	.729177820	.743762550	.778762227	.815408905
2200.0	.741600981	.755753566	.789665243	.825098583
2300.0	.753047361	.766787894	.799667135	.833956212
2900.0	.806256068	.817901874	.845595477	.874226767
3400.0	.836883716	.847183349	.871589874	.896699528
3900.0	.859831303	.869049437	.890836222	.913169194
4400.0	.877596852	.885931230	.905589588	.925684154
4900.0	.891714876	.899315531	.917214496	.935469702
5400.0	.903176109	.910158897	.926581344	.943300111
5900.0	.912646811	.919102744	.934269632	.949686800

## APPENDIX D

### SIMULATOR SOFTWARE

A complete listing of the simulator software is included in this appendix. The software was specifically written for a configuration consisting of a PDP-11/10 minicomputer with 8k memory, a teletype (including a slow paper-tape reader/punch), and the simulator-interface. Also included in the appendix is a listing of PDP-11 instructions, a brief software description, typical teletype I/O, estimates of the processor time requirements for an RVR update, and comments on the program size.

#### D.1 SOFTWARE DESCRIPTION

##### RVR VISIBILITY PROGRAMME

CALCULATES RVR FOR A GIVEN BASE LENGTH, TRANSMISSIVITY OF A TRANSMISSOMETER, RUNWAY LIGHT INTENSITY AND PILOT'S VISUAL LUMINANCE THRESHOLD. USES ALLARD'S AND KOSCHMIEDER'S LAWS IN CALCULATING RVR. VARIOUS PROVISIONS OF THE PROGRAMME ARE AS FOLLOWS:

##### PARAMETER LIST

LIST IS INPUT FROM TTY KB AND CAN BE INPUT ANYTIME DURING THE PROGRAMME FLOW WITH THE HELP OF EXECUTIVE PROGRAMME. LIST IS AS FOLLOWS:

SCALE: SCALE IN INTEGER ARITHMETIC =2\*\*SCALE  
EXITSC: EXIT CRITERIAN IN ITERATIVE ROUTINES IS--  
EXIT IF CHANGE IN RVR<RVR/2\*\*EXITSC  
BASE: BASE LENGTH OF TRANSMISSOMETER, IN FT.  
PRATE: MAX. PULSE RATE. IN PULSES/SEC  
TW: TIME WINDOW, IN SECS, DURING WHICH PULSES ARE COUNTED.  
INMODE: INPUT MODE IN DATA LIST DESCRIBED BELOW.  
INMODE=0: DATA LIST INPUT FROM TTY KB  
INMODE=1: DATA LIST INPUT FROM INTERFACE.

##### DATA LIST

DATA LIST: INTSTY, ET, NPUL1, NPUL2  
IT IS ASSEMBLED FROM TTY OR INTERFACE DEPENDING ON THE FLAG INMODE. ASSEMBLY IS DESCRIBED BELOW.

TTY MODE:

INTSTY: RUNWAY LIGHT INTENSITY ,IN CANDLES  
ET: PILOT'S VISUAL ILLUMINANCE THRESHOLD,MILE-CANDLES.  
NPUL1: PULSE COUNT FOR TR.1.  
NPUL2: PULSE COUNT FOR TR.2.

INTERFACE MODE:

INTSTY:ASSEMBLED BY CHECKING POSITION OF RUNWAY  
LIGHT SETTING SWITCH.  
POS. INTENSITY,CANDLES  
1 400  
2 2.000  
3 10.000  
4 20.000  
ET: ASSEMBLED BY CHECKING B/G ILLUMINANCE  
SWITCH SETTING.  
POS. ET,MILE-CANDLES  
1 2 NIGHT,  
2 26 INTERMEDIATE  
3 260 NORMAL DAY  
4 2600 BRIGHT DAY  
NPUL1: PUPSE COUNT ACCUMULATED DURING TW SECS.  
PULSE RATE DEPENDS ON SWITCH SETTING.  
POS. FREQ. /SEC  
0 10000  
1 100  
2 50  
3 25  
4 12.5  
5 6.25  
6 3.125  
7 1.00  
8 0.50  
9 0.167  
NPUL2: PULSE COUNT FROM PRESET COUNT TRANSMISSOMETER  
RSELC: RUNWAY SELECTOR SWITCH POSITION SETTING FOR  
STROBING CORRESPONDING RVR ON VISUAL DISPLAY.

NOTE--IN INTERFACE MODE A PROVISIN IS MADE IN  
SOFTWARE WHEREBY ONE CAN MAKE CHANGES IN VARIOUS  
SWITCH SETTINGS AT APPROPRIATE RUN TIME.FOR EX.,  
PULSE RATE SWITCH SETTING CAN BE CHANGED IN  
BETWEEN TIME WINDOWS.WHEN & SYMBOL IS PRINTED ON  
TTY.PROGRAMME GOES INTO STALL MODE WAITING FOR  
ANY INPUT FROM TTY KB TO RESUME PROGRAMME  
FLOW.ALSO A RESET SWITCH IS PROVIDED FOR MAKING  
CHANGES IN INTENSITY SWITCH SETTING OR B/G ILLUM-  
INANCE SWITCH SETTING. UNLESS RESET SWITCH IS IN  
POSITION 0, PROGRAMME GOES INTO STALL MODE.

OUTPUT PROVISIONS

- (1)RVR FOR THE TRANSMISSOMETER SELECTED BY RUNWAY SELECTOR SWITCH IS SHOWN ON VISUAL LED DISPLAY EITHER EVERY TIME NEW RVR IS CALCULATED FOR THAT RUNWAY OR RUNWAY SELECTOR SWITCH POSITION IS CHANGED.LATTER ACTION DOES NOT INTERFERE WITH THE CALCULATIONS THAT MAY BE UNDER PROGRESS.
- (2)RVR AND RELEVANT DATA ARE PRINTED ON TTY ONCE EVERY TIME WINDOW.
- (3)A PROVISION IS MADE FOR DIAGNOSTIC PRINTOUT OF INTERMEDIATE RVR ITERATE VALUES DURING ALLARD'S LAW CALCULATIONS. IF FLAG DIAGNOIS OTHER THAN ZERO,FLAG DIAGNO IS ENTERED FROM TTY KB DURING PARAMETER INPUT MODF.



## CALCULATION OF RVR

RVR IS CALCULATED IN INTEGER ARITHMETIC USING ALLARD'S AND KOSCHMIEDER'S LAWS. FOR ALLARD'S LAW PROVISION IS MADE FOR THREE DIFFERENT ITERATIVE ALGORITHMS DEPENDING UPON FLAG MODNR.

MODNR=0	NEWTON-RAPHSON METHOD
=1	MOD. NEWTON-RAPHSON METHOD
=2	HALLEY'S METHOD

NORMALLY MODNR=0. IF OTHER METHODS ARE TO BE USED, CORRESPONDING VALUE FOR MODNR HAS TO BE ENTERED FROM SWITCH REGISTER IN APPROPRIATE MEMORY LOCATION.

## MONITORING PROVISIONS

MONITORING OF THE PROGRAMME CAN BE DONE DURING RUNTIME DIRECTLY BY ISSUING MONITOR COMMANDS FROM TTY KB. THE MONITOR COMMANDS PROVIDE FOR:

-SETTING THE TIME-OF-DAY AND DATE-OF-DAY FROM TTY KB ANYTIME WITHOUT DISRUPTING THE RUNNING OF THE VISIBILITY PROGRAMME. ONCE SO ENTERED BOTH TIME AND DATE ARE UPDATED INTERNALLY AS LONG AS COMPUTER IS RUNNING. ALSO THE TIME AND THE DATE OF THE DAY CAN BE PRINTED ON TTY ANYTIME WITHOUT DISRUPTING THE FLOE OF VISIBILITY PROGRAMME.

-THERE ARE THREE SEPARATE SOFTWARE CHECK-OUT PROVISIONS FOR ALL THE THREE SOFTWARE MODULES .VIZ.

TTY I/O AND EXECUTIVE PACKAGE  
ARITHMETIC PACKAGE  
VISIBILITY PACKAGE

AS EXPLAINED ABOVE .FOR THE LAST PACKAGE. THERE ARE PROVISIONS FOR:

ENTERING THE APPROPRIATE DATA LIST FROM TTY TO SIMULATE INTERFACE INPUTS.

CHANGING THE BASIC PARAMETER LIST FOR RVR SYSTEM FROM TTY.

THREE DIFFERENT ITERATIVE ALGORITHMS FOR RVR COMPUTATION FROM ALLARD'S LAW.

DIAGNOSTIC PRINTOUT TO CHECK THE RVR ITERATES IN THESE ALGORITHMS.

D.2 TYPICAL TELETYPE I/O

Typical teletype I/O is shown in Table D-1. The entries from the keyboard are echoed on the teletype and are underlined in the printout. The printout shows:

- (1) Input data list which selects the scale factor, exit scale factor, diagnostic mode flag, initial RVR guess, transmissometer baselength, maximum pulse rate (for  $t_b = 1$ ), the length of the time window in seconds, and the input mode for the data list.
- (2) Data list inputs from the teletype including the runway-light intensity, visual-illumination threshold, and simulated pulse counts from the transmissometers.
- (3) Diagnostic printout of intermediate values of RVR, function  $f(v)$ , and the increments in RVR during iterative solution of Allard's Law.
- (4) Printout of the input data and the appropriate RVR values.

TABLE D-1. TYPICAL TELETYPE INPUT/OUTPUT PRINTOUT

```

SCALE=10
EXITSC=14, DIAGNO=1, VI=1000 , BASE=60
PRATE=1000, TW=10, INMODE=0

INTSTY=10000, ET=26
NPUL1=7632, NPUL2=9280

V          F(V)          DELV
+01525     -01064        +00525
+01734     -00351        +00209
+01787     -00085        +00053
+01798     -00018        +00011
+01801     -00005        +00003
+01801     -00001        +00000

V          F(V)          DELV
+02377     -01789        +01377
+03829     -01025        +01452
+04592     -00414        +00763
+04841     -00124        +00249
+04904     -00031        +00063
+04920     -00008        +00016
+04924     -00002        +00004
+04926     -00001        +00002
+04926     +00000        +00000

ET         I           NPUL1      RVR1      NPUL2      RVR2
+00026     +10000        +07632    +01801    +09280    +04926

```

### D.3 PDP-11 — INSTRUCTION SET

A brief repertoire of the PDP-11 instructions is reproduced from the PDP-11 handbook.

#### INSTRUCTIONS

Mnemonic	Instruction Operation	OP Code	Condition Codes ZNCV	Timing
<b>DOUBLE OPERAND GROUP: OPR src, dst</b>				
MOV(B)	MOVe (Byte) (src) → (dst)	.1SSDD	✓✓-0	2.3
CMP(B)	CoMPare (Byte) (src) - (dst)	.2SSDD	✓✓✓✓	2.3*
BIT(B)	BIT Test (Byte) (src) ∧ (dst)	.3SSDD	✓✓-0	2.9*
BIC(B)	BIT Clear (Byte) ~(src) ∧ (dst) → (dst)	.4SSDD	✓✓-0	2.9
BIS(B)	BIT Set (Byte) (src) ∨	.5SSDD	✓✓-0	2.3
ADD	ADD (src) + (dst) → (dst)	06SSDD	✓✓✓✓	2.3
SUB	SUBtract (dst) - (src) → (dst)	16SSDD	✓✓✓✓	2.3
<b>CONDITIONAL BRANCHES: Bxx loc</b>				
BR	BRanch (unconditionally) loc → (PC)	0004XX	—	2.6
BNE	Branch if Not Equal (Zero) loc → (PC) if Z = 0	0010XX	—	2.6—
BEQ	Branch if Equal (Zero) loc → (PC) if Z = 1	0014XX	—	2.6—
BGE	Branch if Greater or Equal (Zero) loc → (PC) if N ∨ V = 0	0020XX	—	2.6—
BLT	Branch if Less Than (Zero) loc → (PC) if N ∨ V = 1	0024XX	—	2.6—
BGT	Branch if Greater Than (Zero) loc → (PC) if Z ∨ (N ∨ V) = 0	0030XX	—	2.6—
BLE	Branch if Less Than or Equal (Zero) loc → (PC) if Z ∨ (N ∨ V) = 1	0034XX	—	2.6—
BPL	Branch if Plus loc → (PC) if N = 0	1000XX	—	2.6—
BMI	Branch if Minus loc → (PC) if N = 1	1004XX	—	2.6—
BHI	Branch if Higher loc → (PC) if C ∨ Z = 0	1010XX	—	2.6—
BLOS	Branch if Lower or Same loc → (PC) if C ∨ Z = 1	1014XX	—	2.6—
BVC	Branch if oVerflow Clear loc → (PC) if V = 0	1020XX	—	2.6—
BVS	Branch if oVerflow Set loc → (PC) if V = 1	1024XX	—	2.6—
BCC (or BHIS)	Branch if Carry Clear loc → (PC) if C = 0	1030XX	—	2.6—
BCS (or BLO)	Branch if Carry Set loc → (PC) if C = 1	1034XX	—	2.6—
<b>SUBROUTINE CALL: JSR reg, dst</b>				
JSR	Jump to SubRoutine (dst) → (tmp), (reg) ↓ (PC) → (reg), (tmp) → (PC)	004RDD	—	4.4
<b>SUBROUTINE RETURN: RTS reg</b>				
RTS	ReTurn from Subroutine (reg) → PC, ↑(reg)	00020R	—	3.5

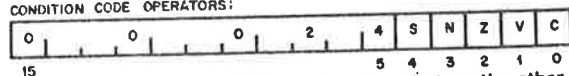
**SINGLE OPERAND GROUP: OPR dst**

CLR(B)	CLeAR (Byte) 0 → (dst)	-050DD	1000	2.3
COM(B)	COMplement (Byte) ~ (dst) → (dst)	-051DD	✓✓00	2.3
INC(B)	INCrement (Byte) (dst) + 1 → (dst)	-052DD	✓✓—✓	2.3
DEC(B)	DECrement (Byte) (dst) - 1 → (dst)	-053DD	✓✓—✓	2.3
NEG(B)	NEGate (Byte) ~ (dst) + 1 → (dst)	-054DD	✓✓✓✓	2.3
ADC(B)	ADd Carry (Byte) (dst) + (C) → (dst)	-055DD	✓✓✓✓	2.3
SBC(B)	SuBtract Carry (Byte) (dst) - (C) → (dst)	-056DD	✓✓✓✓	2.3
TST(B)	TeST (Byte) 0 - (dst)	-057DD	✓✓00	2.3*
ROR(B)	ROtate Right (Byte) rotate right 1 place with C	-060DD	✓✓✓✓	2.3°
ROL(B)	ROtate Left (Byte) rotate left 1 place with C	-061DD	✓✓✓✓	2.3°
ASR(B)	Arithmetic Shift Right (Byte) shift right with sign extension	-062DD	✓✓✓✓	2.3°
ASL(B)	Arithmetic Shift Left (Byte) shift left with lo-order zero	-063DD	✓✓✓✓	2.3°
JMP	JuMP (dst) → (PC)	0001DD	—	1.2
SWAB	SWAp Bytes bytes of a word are exchanged	0003DD	✓✓00	2.3

**CONDITION CODE OPERATORS: OPR**

Condition Code Operators set or clear combinations of condition code bits. Selected bits are set if S = 1 and cleared otherwise. Condition code bits corresponding to bits set as marked in the word below are set or cleared.

CONDITION CODE OPERATORS:



Thus SEC = 000261 sets the C bit and has no effect on the other condition code bits (CLC = 000241 clears the C Bit)

**OPERATE GROUP: OPR**

HALT	HALT processor stops; (RO) and the HALT address in lights	000000	—	1.8
WAIT	WAIT processor releases bus, waits for interrupt	000001	—	1.8
RTI	ReTurn from Interrupt ↑ (PC), ↑ (PS)	000002	••••	4.8
IOT	Input/Output Trap (PS) ↓, (PC) ↓, (20) → (PC), (22) → (PS)	000004	••••	9.3
RESET	RESET an INIT pulse is issued by the CP	000005	—	20 ms
EMT	EMulator Trap (PS) ↓, (PC) ↓, (30) → (PC), (32) → (PS)	104000—104377	••••	9.3
TRAP	TRAP (PS) ↓, (PC) ↓, (34) → (PC), (36) → (PS)	104400—104777	••••	9.3

**NOTATION:**

- for order codes
  - word/byte bit, set for byte ( ; 100000)
  - SS—source field,
  - DD—destination field
  - XX—offset (8 bit)
- for operations
  - ^ and,
  - v or,
  - ~ not,
  - ( ) contents of,
  - ∨ XOR
  - ↓ "is pushed onto the processor stack"
  - ↑ "the contents of the top of the processor stack is popped and becomes"
  - "becomes"
- for timing
  - 0.4 μs less if not register mode
  - 0.9 μs less if conditions for branch not met
  - 1.2 μs more if addressing odd byte
  - (0.6 μs additional in addressing odd bytes otherwise)
- for condition codes
  - ✓ set conditionally
  - not affected
  - 0 cleared
  - 1 set

#### D.4 RVR UPDATE TIME REQUIREMENTS

Table D-2 gives the estimates of processor time requirements for different elements of computations involved in an RVR update. Estimates are given for two cases when integer and floating-point arithmetic is employed in the calculations. The integer and floating-point multiply and divide capability is assumed to be supplied by software rather than by hardware. Thus, it may be seen that for integer and floating-point arithmetic calculations the processor time required to update RVR for n transmissometers is less than or equal to 0.1 seconds and 0.3 seconds, respectively. The time involved in the data input operations and in the output of RVR to the display is very small compared to the time involved in the RVR-update computations. In the above, estimate for the time involved in the teletype operations, if any, is not included. The teletype speed and the number of characters in the output list determine the time involved.

TABLE D-2. ESTIMATES OF PROCESSOR TIME REQUIREMENTS FOR RVR UPDATE

Step		Time (s)	
No.	Description	Integer Arithmetic	Floating-Point Arithmetic
1	Allard's Law using N-R iteration:		
	(1) per iteration	0.01	0.03
	(2) for ten maximum iterations	0.10	0.30
2	Koschmieder's Law	0.003	0.011
3	RVR Update		
	(1) average (five iterations)	<0.05	0.15
	(2) maximum	0.1	0.30

#### D.5 PROGRAM SIZE

Table D-3 shows the size of the four basic programs that form part of the simulator software. VISIB updates and displays RVR based on the data from various sensors; ARITH provides the necessary integer arithmetic routines; TTYIO handles the teletype I/O operations and the related format conversion functions; and EXEC provides the control by

keyboard commands. The total size of the program is 1754 memory locations. The routines in ARITH and TTYIO were specifically written to provide the necessary functions keeping the memory size requirements to a minimum. For a basic 4k memory configuration, room exists for further expansion of the program by about 1700 words, leaving aside about 600 words for interrupt vectors, stack requirements, and for the loaders.

TABLE D-3. SIZE OF THE PROGRAM SIMULATOR SOFTWARE

Program	Memory Size in Words	Comments
VISIB	887	RVR computations, I/O
ARITH	314	Arithmetic routines
TTYIO	388	Teletype I/O
EXEC	165	Executive
TOTAL	<u>1754</u>	

## D.6 SIMULATOR SOFTWARE

```

:
:RVR VISIBILITY PROGRAMME
:
:SUBROUTINES:
:VISIB: MAIN PROGRAMME
:CALC:
:ITER:
:   ABOVE TWO SUPPLY NEW RVR VALUES
:FOLLOWING SERVICE OUTPUT PROVISIONS
:OUTPUT:   GENERAL ROUTINE
:OUTATC:   FOR VISUAL DISPLAY
:OUTREC:   TO TTY FOR RECORD
:STROBE:
:DISPLA:
:OUTSTR:
:   ABOVE THREE ARE AUXILLIARY OUTPUT ROUTINES
:FOLLOWING SERVICE INPUT PROVISIONS
:INPUT: GENERAL ROUTINE
:TTYDAT:DATA FROM TTY
:INTFAC:DATA FROM INTERFACE USING SUBROUTINES-
:   TRIN:  DATA FROM TRANSMISSOMETER COUNTERS
:
:   RINTIN: RUNWAY LIGHT SETTING
:   BGILIN: B/G ILLUMINANCE AND RUNWAY SELCTOR
:           SETTING
:TTYPAR:PARAMETER LIST FROM TTY
:
:
:LKSERV:KEEPS TRACK OF TIME(HR,MIN,SEC,ARCSEC),DAY
:   (DAY,MONTH,YEAR).START AND STOP PULSE COUNTERS,
:   AND CHECKS RUNWAY SELECTOR SWITCH SETTING EVERY
:   ONE SECOND.
:
:
:
:
:RUNWAY VISUAL RANGE PROGRAMME
000000 R0=%0
000001 R1=%1
000002 R2=%2
000003 R3=%3
000004 R4=%4
000005 R5=%5
000006 SP=%6
000007 PC=%7
.GLOBL LSHIFT, SCALE, DIV, MULT, SCALOG, RSHIFT, MONDAY
.GLOBL VISIB, THR, TMIN, TSEC, TDAY, TMON, TYEAR, MONDET, LKSERV
.GLOBL GETCHR, GETNUM, ICA, ICABUF, INIT, MESS, PUTCHR, INBUF
.TITLE VISIB
000000 .ASECT
```

PAGE 001

005600 . =5600

```
005600 005067 CALC: CLR ITRANS
002750
005604 016700 MOV INTSTY,R0
002674
005610 004567 JSR R5,SCALOG
000000
005614 010002 MOV R0,R2 ;SAVE
005616 016700 MOV ET,R0
002664
005622 004567 JSR R5,SCALOG ;LOG(ET)*2**S
000000
005626 160200 SUB R2,R0 ;2**S*(LOG(ET)-LOG(INTSTY))
005630 006200 ASR R0 ;DIVIDE BY 2
005632 166700 SUB L5280,R0
002764
005636 010067 MOV R0,CAIK
002732
005642 016700 CALBACK: MOV ITRANS,R0
002706
005646 006300 ASL R0
005650 016000 MOV NPUL(R0),R0
010606
005654 004567 JSR R5,SCALOG ;2**S*LOG(NB)
000000
005660 166700 SUB LOGNBM,R0 ;2**S*LOG(TB)
002712
005664 010067 MOV R0,CAIB
002702
005670 016700 MOV ITRANS,R0
002660
005674 006300 ASL R0
005676 016067 MOV VSTORE(R0),V
010602
002674
005704 016767 MOV VINIT,V
002702
002666
005712 004567 JSR R5,ITER ;SUPPLIES NEW RVR
000032
005716 016700 MOV ITRANS,R0
002632
005722 006300 ASL R0
005724 016760 MOV V,VSTORE(R0) ;STORE NEW RVR
002650
010602
005732 005267 INC ITRANS
002616
005736 026727 CMP ITRANS,#NTRANS
002612
000002
005744 002736 BLT CALBACK ;NO.GO BACK
005746 000205 RTS R5
```



```

;
;SUBROUTINE ITER
;SUPPLIES NEW RVR
;
005750 012767 ITER:MOV #1,NOITER
          000001
          002600
005756 004567 JSR R5,DPRNT1 ;DIAGNOSTIC PRINT ROUTINE
          000414

005762 016700 ITERLUP: MOV V,R0
          002612
005766 004567* JSR R5,SCALOG ;2**S*LOG(V)
          000000
005772 010067 MOV R0,ITIV ;STORE
          002572
005776 016700 MOV V,R0
          002576

006002 016702 MOV CAIB,R2 ;V*LOG(TB)*2**S
          002564
006006 004567* JSR R5,MULT
          000000
006012 016702 MOV BASE,R2
          002464
006016 006302 ASL R2

006020 004567* JSR R5,DIV ;R1=V*LOG(TB)*2**S/(BASE*2)
          000000

006024 160167 SUB R1,ITIV
          002540
006030 066767 ADD CAIK,ITIV ;ITIV=CAIK+ITIV-R1
          002540
          002532
006036 015700 MOV MAGSCA,R0 ;MAGSCA=2**S
          002556
006042 160100 SUB R1,R0 ;2**S-ITIV
006044 010067 MOV R0,DUM ;DUM=2**S-ITIV
          002516
006050 005767 TST MODNR
          002554
006054 001433 BEQ ITER10 ; MODNR IS A FLAG TO DECIDE IF MOD.
;N-R METHOD IS TO BE USED
;FOLLOWING CALCULATIONS ARE FOR MOD. N-R METHOD
;
006056 016700 MOV MAGSCA,R0 ;2**S

          002536
006062 016702 MOV ITIV,R2 ;F(V)*2**S
          002502
006066 004567* JSR R5,MULT
          000000

```

PAGE 003

```
006072 016702      MOV DUM,R2      ;F'(V)*V*2**S
          002470
006076 004567*     JSR R5,DIV      ;2**S*F*(2**S)/(2**S*F'*V)=R1
          000000
006102 026727      CMP MODNR,#1    ;MODNR=1:MOD. N-R METHOD
          002522
          000001
                                ;
                                ;=2:HALLEY'S METHOD
006110 001401      BEQ ITER11
006112 006201      ASR R1          ;DIVIDE BY 2

006114 016700 ITER11: MOV DUM,R0
          002446
006120 006200      ASR R0
006122 020100      CMP R1,R0      ;R1<DUM/2?
006124 003004      BGT ITER8     ;IF NOT JUMP
006126 005400      NEG R0
006130 020100      CMP R1,R0      ;R1>-DUM/2?
006132 002401      BLT ITER8     ;IF NOT JUMP
006134 000401      BR ITER9
006136 010001 ITER8: MOV R0,R1      ;SET THE APPROPRIATE
                                ;VALUE FOR R1
006140 060167 ITER9: ADD R1,DUM     ;REVISE DUM
          002422
                                ;
                                ;END OF SPECIAL CALCULATIONS FOR MOD.
                                ;N-R METHOD

006144 016700 ITER10: MOV V,R0
          002430
006150 016702      MOV ITIV,R2
          002414

006154 004567*     JSR R5,MULT     ;(R0,R1)=V*ITIV
          000000
006160 016702      MOV DUM,R2
          002402
006164 004567*     JSR R5,DIV      ;R1=DELV=V*ITIV/DUM
          000000

006170 005701      TST R1
006172 016700      MOV V,R0
          002402
006176 160100      SUB R1,R0      ;VNEW=V-DELV
006200 020027      CMP R0,#VMAX   ;VNEW<VMAX?
          013560
006204 003402      BLE ITER1     ;YES,SKIP
006206 012700      MOV #VMAX,R0   ;VNEW=VMAX

          013560
006212 020027 ITER1: CMP R0,#VMIN
          000062
006216 002002      BGE ITER2
```

PAGE 004

```
006220 012700      MOV #VMIN,R0      ;VNEW=VMIN IF VNEW<VMIN
          000062
006224 010067 ITER2: MOV R0,V ;V=VNEW
          002350
006230 016702      MOV MEXIT,R2
          002372
006234 006200 ITER3:ASR R0
006236 005302      DEC R2
006240 001375      BNE ITER3
006242 005700      TST R0
006244 100002      BPL ITER4
```

PAGE 005

```
006246 012700      MOV#1,R0      ;EXIT= 1, IF V/2**NEXITIS ZERO OR
000001                                     ; LESS THAN ZERO

006252 005701 ITER4: TST R1
006254 100001      BPL ITER5
006256 005401      NEG R1
006260 004567 ITER5: JSR R5,DPRNT2 ;DIAGNOSTIC PRINT
000132

006264 020100      CMP R1,R0      ;DELV<V/2**SCALE?

006266 003407      BLE ITEREXIT ;YES,DONE,EXIT
006270 026727      CMP NOITER,#ITERMAX ;NOITER .GT. ITERMAX?
002262
000012

006276 003021      BGT ITERERR  ;YES,PRINT MESSAGE,EXIT
006300 005267      INC NOITER
002252

006304 000626      BR ITERLUP   ;LOOP BACK
006306 016700 ITEREXIT:MOV BALEPS,R0
002250
006312 016701      MOV BALEPS+2,R1
002246
006316 016702      MOV CAIB,R2
002250
006322 004567*     JSR R5,DIV
000000
006326 020167      CMP R1,V
002246
006332 002402      BLT ITER6
006334 010167      MOV R1,V      ;NEW RVR IS AS CALCULATED BY KOSCHMEIDER
002240                                     ;LAW, SINCE IT IS GREATER THAN THAT CALC
                                          ;ULATED BY ALLARD'S LAW.

006340 000205 ITER6: RTS R5

006342 010446 ITERERR: MOV R4,-(SP)
006344 012704      MOV #ITERMES,R4
006360
006350 004567*     JSR R5,MESS
000000
006354 012604      MOV (SP)+,R4
006356 000753      BR ITEREXIT
006360 015 ITERMES:.BYTE 15,12
006361 012
006362 115      .ASCII /MAX ITER/
006363 101
006364 130

006365 040
006366 111
006367 124
006370 105
```

```

006371 122
006372 015 ITCRLF: .BYTE 15,12,0
006373 012
006374 000
      006376 .EVEN
      ;
      ;DIAGNOSTIC PRINTOUT ROUTINE
      ;
006376 005767 DPRNT1: TST DIAGNO      ;ANY DIAGNOSTIC?
      002212
006402 001404      BEQ DPR1
006404 012704      MOV #ITMESS,R4
      006446
006410 004567      JSR R5,MESS
      000000
006414 000205 DPR1:  RTS R5
      ;
006416 005767 DPRNT2: TST DIAGNO      ;ANY DIAGNOSTICS?
      002172
006422 001410      BEQ DPR2
006424 010167      MOV R1,DUM
      002136
006430 004567      JSR R5,ICABUF      ;SET UP OUTPUT LIST
      000000
006434 000003      .WORD 3
006436 010600      .WORD V,ITIV,DUM
006440 010570
006442 010566
006444 000205 DPR2:  RTS R5
      ;
      ;
006446 015 ITMESS: .BYTE 15,12
006447 012
006450 126      .ASCII /V      F(V)
006451 040
006452 040
006453 040
006454 040
006455 040
006456 040
006457 040
006460 040
006461 040
006462 106
006463 050
006464 126
006465 051
006466 040
006467 040
006470 040
006471 040
006472 040

006473 040
006474 104      .ASCII /DELV/
006475 105
006476 114

```

PAGE 007

```
006477 126
006500 015 .BYTE 15,12,0
006501 012
006502 000
006504 .EVEN
;
;
;
;SUBROUTINE OUTPUT
;
006504 010246 OUTPUT:MOV R2,-(SP)
006506 010346 MOV R3,-(SP)
006510 010446 MOV R4,-(SP)
006512 004567 JSR R5,OUTATC ;AIR TRAFFIC CONTROL DISPLAY
000014
006516 004567 JSR R5,OUTREC ;RECORDING ROUTINE
000024
006522 012604 MOV (SP)+,R4
006524 012603 MOV (SP)+,R3
006526 012602 MOV (SP)+,R2
006530 000205 RTS R5
;
;AIR TRAFFIC CONTROL DISPLAY ROUTINE
;
```

```
006532 005767 OUTATC: TST INMODE
001754
006536 001402 BEQ OUTA1
006540 004567 JSR R5,OUTSTR
001542
006544 000205 OUTA1: RTS R5
;
;RECORDS INFORMATION-HERE ON TTY
;
```

```
006546 012704 OUTREC: MOV #OUTMES,R4
006502
```

```
006552 004567 JSR R5,MESS
000000
006556 004567 JSR R5,ICABUF
000000
006562 000006 .WORD 6.,ET,INTSTY,NPUL,VSTORE,NPUL+2
006564 010506
006566 010504
006570 010606
006572 010602
006574 010610
006576 010604 .WORD VSTORE+2
006600 000205 RTS R5
;
```

```
006602 015 OUTMES: .BYTE 15,12
006603 012
006604 105 .ASCII /ET I NPUL1
```

PAGE 010

006605 124  
006606 040  
006607 040  
006610 040  
006611 040  
006612 040  
006613 040  
006614 040  
006615 040  
006616 111  
006617 040  
006620 040  
006621 040  
006622 040  
006623 040  
006624 040  
006625 040  
006626 040  
006627 040  
006630 116  
006631 120  
006632 125  
006633 114  
006634 061  
006635 040  
006636 040  
006637 040  
006640 040  
006641 040  
006642 122  
006643 126  
006644 122  
006645 061  
006646 040  
006647 040  
006650 040  
006651 040  
006652 040  
006653 040  
006654 116  
006655 120  
006656 125  
006657 114  
006660 062  
006661 040  
006662 040  
006663 040  
006664 040  
006665 040  
006666 122  
006667 126  
006670 122

.ASCII /RVR1 NPUL2 RVR2/

006671 062  
006672 015  
006673 012  
006674 000

.BYTE 15,12,0

PAGE 011

```
006676 .EVEN

;ROUTINE SERVES LINE CLOCK INTERRUPT
;KEEPS TRACK OF TIME,AND DATE
;STARTS AND STOPS PULSE COUNTERS

;CHECKS RUNWAY SELECTOR SWITCH EVERY ONE SECOND
;;
;
006676 005767 LKSERV:TST CTRFLG
001620

;TIME WINDOW TO BE INITIATED?
006702 001411 BEQ LK1

006704 016767 MOV TW,TWCTR
001620
001620

;TW=T*60,T=TIME WINDOW IN SECS
006712 110037 MOVB R0,#164000
164000

;ENABLE AND RESET C/T COUNTER TO
;ZERO
006716 005067 CLR CTRFLG
001600

;CLEAR FLAGS
006722 005067 CLR INFLAG
001576

006726 005367 LK1: DEC TWCTR
001600
006732 001004 BNE LK2
006734 110037 MOVB R0,#164001
164001

;DIENABLE C/T COUNTER

006740 005267 INC INFLAG
001560

;FLAG TO SIGNIFY READING DONE
006744 005267 LK2: INC TARSEC
001564
006750 026727 CMP TARSEC,#60.
001560
000074
006756 001401 BEQ LK3
006760 000002 RTI

;ENTERED EVERY 1 SECOND
006762 005067 LK3: CLR TARSEC
001546
006766 005267 INC TSEC

001544
006772 026727 CMP TSEC,#60.
001540
000074
```



PAGE 012

```
007000 002452      BLT LK9
007002 005067      CLR TSEC
          001530
007006 005267      INC TMIN
          001526
007012 026727      CMP TMIN,#60.
          001522
          000074
007020 002442      BLT LK9
007022 005067      CLR TMIN
          001512
007026 005267      INC THR
          001510
007032 026727      CMP THR,#24.
          001504
          000030
007040 002432      BLT LK9
          001474
007042 005067      CLR THR
          001474
007046 005267      INC TDAY
          001472
007052 026767      CMP TDAY,MONDAY
          001466
          001466
007060 002422      BLT LK9
007062 012767      MOV #1,TDAY
          000001
          001454
007070 005267      INC TMON
          001454
007074 026727      CMP TMON,#13.
          001450
          000015
007102 002405      BLT LK8
007104 012767      MOV #1,TMON
          000001
          001436
          001434
007112 005267      INC TYEAR
          001434
007116 016646 LK8:  MOV 2(SP),-(SP)
          000002
007122 012746      MOV #MONDET,-(SP)
          000000
007126 005767 LK9:  TST INMODE
          001360
          001360
007132 001001      BNE LK4
007134 000002      RTI
          001360
007136 010046 LK4:  MOV R0,-(SP)
007140 010146      MOV R1,-(SP)
007142 016700      MOV RSELECT,R0
          001360
```

```

                                ; INTERFACE MODE
007146 005067 CLR RSELECT
                                001354
007152 113701 MOVW @#164004,R1
                                164004

                                ; WORD CONTAINING R/SETTING BIT
007156 106001 RORB R1
007160 005567 ADC RSELECT
                                001342
007164 012601 LK5:  MOV (SP)+,R1 ;CHECK IF
007166 026700 CMP RSELECT,R0 ;STATUS OF FLAG
                                001334
007172 001002 BNE LK7 ;IS CHANGED
007174 012600 MOV (SP)+,R0 ;IF NOT RETURN
007176 000002 RTI
007200 012600 LK7:  MOV (SP)+,R0
007202 016646 MOV 2(SP),-(SP) ;MOVE PROCESSOR STATUS OF INTERRUPTED
                                000002

                                ; PROGRAMME TO STACK
007206 012746 MOV #STROBE,-(SP)
                                010300

                                ; STATUS OF RFLAG
007212 000002 RTI ;PUSH ADDRESS OF STROBE
                                ;ROUTINE AND OLD PSWORD
                                ;ON THE STACK IN THAT ORDER

; INPUT ROUTINE FOR PROGRAMME
;
007214 005767 INPUT: TST INMODE ;CHECK INPUT MODE
                                001272
007220 001403 BEQ INTTY ;BRANCH IF TTY MODE
007222 004567 JSR R5,INTFAC ;INTERFACE MODE
                                000602
007226 000205 RTS R5
007230 004567 INTTY: JSR R5,TTYDAT ;DATA COMES FROM TTY
                                000002
007234 000205 RTS R5
                                ; IN THE ARRAY
;
;
;SUBROUTINE TTYDAT
;TAKES DATA FROM TTY
007236 004567 TTYDAT: JSR R5,SPACE
                                000532
007242 000001 .WORD 1

007244 004567 JSR R5,INBUF
                                000000
007250 000004 .WORD 4
007252 007274 .WORD MWORD6,INTSTY ;INTENSITY OF RUNWAY LIGHTS
007254 010504

007256 007304 .WORD MWORD7,ET ;ILLUMINANCE THRESHOLD
007260 010506
007262 007310 .WORD MWORD8,NPUL ;PULSE COUNT FOR C/T TR
007264 010606

```

```

007266 007320      .WORD MWORD9, NPUL+2      :PULSE COUNT FOR BCD INPUT TR
007270 010610

007272 000205 RTS R5
;
007274      111 MWORD6: .ASCII /INTSTY=/
007275      116
007276      124
007277      123
007300      124
007301      131
007302      075
007303      000 .BYTE 0
007304      105 MWORD7: .ASCII /ET=/
007305      124
007306      075
007307      000 .BYTE 0
007310      116 MWORD8: .ASCII /NPUL1=/
007311      120
007312      125
007313      114
007314      061
007315      075
007316      000 .BYTE 0
007320      116 MWORD9: .ASCII /NPUL2=/
007321      120
007322      125
007323      114
007324      062
007325      075
007326      000 .BYTE 0
007330      .EVEN
;
;
;SUBROUTINE TTYPAR
; SUPPLIES PARAMETERS-BASIC;
007330 004567 TTYPAR: JSR R5,SPACE
000440
007334 000005      .WORD 5

007336 004567*      JSR R5, INBUF
000000

007342 000003      .WORD 3
007344 007744      .WORD NWORD1.SCALE
007346 010616
007350 007754      .WORD NWORD2.MEXIT
007352 010626
007354 007764      .WORD NWORD3.DIAGNO

007356 010614

```

PAGE 015

007360 012700  
000001

MOV #1,R0

007364 016701  
001226

MOV SCALE,R1

007370 004567  
000000

JSR R5,LSHIFT

007374 010067  
001220

MOV R0,MAGSCA ;MAGSCA=2\*\*SCALE

007400 012700  
012240

MOV #5280.,R0 ;MILE

007404 004567  
000000

JSR R5,SCALOG

007410 010067  
001206

MOV R0,L5280 ;L5280=LOG10(5280)\*2\*\*SCALE

007414 012700  
050223

MOV #20627.,R0

007420 012701  
000016

MOV #14.,R1 ;-20627.=LOG10(EPS=.055)\*2\*\*15

007424 166701  
001166

SUB SCALE,R1

007430 004567  
000000

JSR R5,RSHIFT ;RIGHT SHIFT ROUTINE

007434 005400

NEG R0

007436 010067  
001162

MOV R0,LEPS ;LEPS=LOG10(EPS)\*2\*\*SCALE

007442 004567 TTYPA4: JSR R5,INBUF  
000000

007446 000005

.WORD 5

007450 007706

.WORD MWORD1,VINIT

007452 010612

.WORD MWORD2,BASE

007456 010502

.WORD MWORD3,PRATE

007460 007720

007462 010510

.WORD MWORD4,TW

007464 007730

007466 010530

.WORD MWORD5,INMODE

007470 007734

007472 010512

007474 016767  
001112  
001100

MOV VINIT,VSTORE

007502 016767  
001104  
001074

MOV VINIT,VSTORE+2

007510 016702

MOV PRATE,R2

000774

PAGE 016

```
007514 016700      MOV TW,R0
          001010
007520 004567*    JSR R5,MULT
          000000
007524 010100      MOV R1,R0      ;MAX PULSE COUNT
007526 004567*    JSR R5,SCALOG
          000000
007532 010067      MOV R0,LOGNBM
          001040
007536 016700      MOV TW,R0
          000766
007542 012702      MOV #60.,R2
          000074
007546 004567      JSR R5,MULT
          000000
007552 010167      MOV R1,TW      ;NOW TW=TW*60
          000752
007556 016700      MOV BASE,R0
          000720
007562 016702      MOV LEPS,R2
          001036
007566 004567      JSR R5,MULT
          000000
007572 010067      MOV R0,BALEPS
          000764
007576 010167      MOV R1,BALEPS+2      ;STORE BASE*LOG10(EPS=.05)*2**S
          000762      ;AT BALEPS IN DOUBLE PRECISION

007602 005267      INC FSTIME
          000712
007606 012767 TTYPA3: MOV #NINPUT,NCOUNT
          000005
          000702

007614 005767 TTYPA1: TST NCOUNT
          000676

007620 001401      BEQ PBACK
007622 000205      RTS R5
007624 012704 PBACK: MOV #PARMES,R4
          007656
007630 004567*    JSR R5,MESS
          000000
007634 004567*    JSR R5.GETCHR
          000000
007640 120027      CMPB R0,#131
          000131

007644 001676      BEQ TTYPA4
007646 000757      BR TTYPA3

;
007650 005367 TTYPA2: DEC NCOUNT
          000642
007654 000757      BR TTYPA1
```

```

007656 015 PARMES: .BYTE 15,12
007657 012
007660 120 .ASCII /PARAMETER CHANGE ?/
007661 101
007662 122
007663 101
007664 115
007665 105
007666 124
007667 105
007670 122
007671 040
007672 103
007673 110
007674 101
007675 116
007676 107
007677 105
007700 040
007701 077
007702 015 .BYTE 15,12,0
007703 012
007704 000
007706 .EVEN
007706 126 MWORD1: .ASCII /VI=/
007707 111
007710 075
007711 000 .BYTE 0
007712 .EVEN
007712 102 MWORD2: .ASCII /BASE=/
007713 101
007714 123
007715 105
007716 075
007717 000 .BYTE 0
007720 .EVEN
007720 120 MWORD3: .ASCII /PRATE=/
007721 122
007722 101
007723 124
007724 105
007725 075
007726 000 .BYTE 0
007730 .EVEN
007730 124 MWORD4: .ASCII /TW=/
007731 127
007732 075
007733 000 .BYTE 0
007734 .EVEN
007734 111 MWORDS: .ASCII /INMODE=/
007735 116

007736 115
007737 117
007740 104
007741 105

```

PAGE 020

```
007742 075
007743 000 .BYTE 0
007744 .EVEN

007744 123 NWORD1: .ASCII /SCALE=/
007745 103
007746 101
007747 114
007750 105
007751 075
007752 000 .BYTE 0
007754 .EVEN
007754 105 NWORD2: .ASCII /EXITSC=/
007755 130
007756 111
007757 124
007760 123
007761 103
007762 075
007763 000 .BYTE 0

007764 104 NWORD3: .ASCII /DIAGNO=/
007765 111
007766 101
007767 107
007770 116
007771 117
007772 075
007773 000 .BYTE 0
007774 .EVEN
007774 .EVEN
;
;
; THIS SUBROUTINE MAKES TELEPRINTER
; ADVANCE BY ONE LINE
;
007774 010146 SPACE: MOV R1,-(SP)
007776 010446 MOV R4,-(SP)
010000 012501 MOV (R5)+,R1
010002 012704 SPABAC: MOV #SPCRLF,R4
010024
010006 004567 JSR R5,MESS
000000
010012 005301 DEC R1
010014 001372 BNE SPABAC
010016 012604 MOV (SP)+,R4
010020 012601 MOV (SP)+,R1

010022 000205 RTS R5
010024 015 SPCRLF: .BYTE 15,12,0
010025 012
010026 000

010030 .EVEN
;
;
```

```

      ;:SUBROUTINE INTFAC
      :ACCEPTS DATA FROM INTERFACE
      :
010030 004567 INTFAC:JSR R5,TRIN      ;INPUTS FROM TRANS. COUNTERS
      000012
010034 004567      JSR R5,RINTIN    ;RUNWAY LIGHT INTENSITY ROUTINE
      000112
010040 004567      JSR R5,BGILIN   ;BACKGROUND ILLUMINANCE AND RUNWAY
      000150
      ;SELCOR SWITCH POSITION.

010044 000205      RTS R5
      :
      :
      ;TRANSMISSOMETER COUNT INPUT ROUTINE
010046 005767 TRIN: TST FSTIME     ;FIRST TIME WINDOW
      000446
010052 001407      BEQ IOD3       ;IF NOT JUMP
010054 005067      CLR FSTIME     ;CLEAR FLAG
      000440
010060 005267      INC CTRFLG     ;SIGNAL TO INTERRUPT ROUTINE TO START
      000436
      ; TIME WINDOW
010064 005767 IOD4: TST CTRFLG    ;HAS IT STARTED?
      000432
010070 001375      BNE IOD4       ;IF NOT WAIT
010072 005767 IOD3: TST INFLAG    ;TIME WINDOW OVER?
      000426
010076 001775      BEQ IOD3       ;IF NOT WAIT
010100 013767      MOV @#164000, NPUL ;READ COUNT DURING TIME WINDOW
      164000
      000500
      ;
010106 112700      MOVB #046,R0    ;ASCII CODE FOR &
      000046
010112 004567*     JSR R5,PUTCHR   ;PUT IT ON KB
      000000
010116 004567*     JSR R5,GETCHR   ;PROGRAMME STALLS HERE TILLIT SEES
      000000
      ;ANY INPUT FROM KB. DURING THIS STALL
      ;PHASE, SWITCH SETTINGS ON INTERFACE
      ;MAY BE CHANGED.

010122 005267      INC CTRFLG
      000374
010126 005767 IOD1: TST CTRFLG    ;HAS THE TIME WINDOW BEEN STARTED?
      000370
010132 001375      BNE IOD1       ;IF NOT WAIT
      :
010134 005737 IOD7: TST @#164002   ;WORD 2 READY?
      164002
010140 100775      BMI IOD7
      :

010142 013767      MOV @#164002, NPUL+2
      164002
      000440
010150 000205      RTS R5

```



```

;
;RUNWAY LIGHT INTENSITY INPUT ROUTINE
;
010152 005737 RINTIN: TST @#164004 ;CHECK IF WORD 3 READY?
          164004
010156 100775      BMI RINTIN ;NO.WAIT
010160 113700      MOV B @#164005,R0 ;HIGHER BYTE
          164005
010164 012701      MOV #3,R1 ;COUNTER
          000003
010170 106200 IOD9: ASRB R0
010172 103002      BCC IOD10
010174 005301      DEC R1
010176 000774      BR IOD9
010200 006301 IOD10: ASL R1
010202 016167      MOV INTTBL(R1),INTSTY
          010260
          000274
010210 000205      RTS R5
;
;ROUTINE SUPPLIES:(1)VISUAL ILLUMINANCE THRESHOLD OF PILOT
;                  (2)RUNWAY SELECTOR SWITCH SETTING -FOR RVR
;                  DISPLAY
010212 010332
;

010214 005067 BGILIN: CLR RSELECT
          000306
010220 113700      MOV B @#164004,R0
          164004
010224 106200      ASRB R0
010226 005567      ADC RSELECT
          000274
010232 106300      ASLB R0
010234 005001      CLR R1
010236 106300 IOD11: ASLB R0
010240 103002      BCC IOD12
010242 005201      INC R1
010244 000774      BR IOD11
010246 006301 IOD12: ASL R1
010250 016167      MOV ETTBL(R1),ET
          010270
          000230
010256 000205      RTS R5

010260 000620 INTTBL: .WORD 400.,2000.,10000.,20000.
010262 003720
010264 023420
010266 047040
010270 000002 ETTBL: .WORD 2.26.,260.,2600.
010272 000032

010274 000404
010276 005050
;
;STROBE ROUTINE

```

```

;SERVICES TIME INTERRUPT ROUTINE LKSERV AT PROGRAMME
;INTERRUPT LEVEL
010300 004567 STROBE: JSR R5,OUTSTR
          000002
010304 000002          RTI
;
;
;OUTSTR ROUTINE STROBES THE RVR FOR
;THE RUNWAY CORRESPONDING TO THE ONE
;REQUESTED BY THE RUNWAY SELECTOR SWITCH
;
;
;
010306 010046 OUTSTR: MOV R0,-(SP)
010310 016700          MOV RSELECT,R0
          000212
010314 006300          ASL R0

010316 016000          MOV VSTORE(R0),R0
          010602
010322 004567          JSR R5,DISPLA
          000004
010326 012600          MOV (SP)+,R0
010330 000205          RTS R5

;
;
;
;DISPLA ROUTINE
;STROBES NO IN R0 TO DISPLAY
;IF NO IS GREATER THAN 9998. ,9998. IS DISPLAYED
;IF NI IS - , 9999 IS DISPLAYED
;
010332 010146 DISPLA: MOV R1,-(SP)
010334 010246          MOV R2,-(SP)
010336 010346          MOV R3,-(SP)
010340 010446          MOV R4,-(SP)
010342 020027          CMP R0,#9998.
          023416
010346 003042          BGT DISPL3
010350 005700          TST R0
010352 100443          BMI DISERR
010354 012702 DISPL4: MOV #BUFFIC,R2
          010470
010360 004567*          JSR R5,ICA          ;BINARY TO ASCII
          000000
010364 012704          MOV #BUFFIC+2,R4          ;SKIP SIGN BYTE AND LEADING DIGI
          010472
;
;          BYTE
010370 012703          MOV #2,R3

          000002
010374 112400 )ISPL1: MOVB (R4)+,R0          ;HIGHER DIGIT
010376 162700          SUB #60,R0          ;CONVERT TO BINARY
          000060

```

PAGE 024

```
010402 110001      MOVB R0,R1
010404 112400      MOVB (R4)+,R0      ;GET NEXT LOWER DIGIT
010406 162700      SUB #60,R0          ;CONVERT TO BINARY
                   000060
010412 006300      ASL R0
010414 006300      ASL R0
010416 006300      ASL R0
010420 006300      ASL R0      ;SHIFT LEFT 4 TIMES
                   ;THE LOWER DIGIT IS NOW IN RIGHT HALF OF LOW BYTE OF R0
010422 060100      ADD R1,R0          ; FORM A BYTE OF TWO DIGITS
010424 005303      DEC R3
010426 001403      BEQ DISPL2       ;JUMP ON SECOND PASS
010430 110037      MOVB R0,#164006      ;ON FIRST PASS STROBE TOP TWO
                   164006
                   ;DIGITS TO DISPLAY
010434 000757      BR DISPL1
010436 110037      DISPL2: MOVB R0,#164007
                   164007
                   ;PASS TWO LOWER DIGITS ARE STROBED TO DISPLAY
010442 012604      MOV (SP)+,R4
010444 012603      MOV (SP)+,R3
010446 012602      MOV (SP)+,R2
010450 012601      MOV (SP)+,R1
010452 000205      RTS R5
010454 012700      DISPL3:MOV #9998,,R0      ;LIMIT NO TO 9998.
                   023416
010460 000735      BR DISPL4
010462 012700      DISERR: MOV #9999. ,R0      ;IF NO IS - ,DISPLAY 99999
                   023417
010466 000732      BR DISPL4
010502 010502      BUFFIC: .=.+10.
                   ;
                   ;.
                   ;CONSTANTS FOR VISIB PROGRAMME
                   ;
000002      NTRANS=2      ;NO OF TRANS
000004      NTRDBL=4      ;NTRANS*2

000012      ITERMAX=10.   ;MAXIMUM NO OF ITERATIONS ALLOWED

013560      VMAX=6000.    ;MAXIMUM RVR LIMIT TO
000062      VMIN=50.     ;MINIMUM RVR LIMIT

000005      NINPUT=5      ;TOTAL NO OF DATA INPUTS BEFORE PARAMETE
                   ;CHANGE CAN BE REQUESTED
```

VARIABLES FOR VISIBILITY PROGRAMME

```
010502 000000      BASE:  .WORD 0
010504 000000      INTSTY: .WORD 0
010506 000000      ET:    .WORD 0
010510 000000      PRATE:  .WORD 0
```

```

010512 000000 INMODE: .WORD 0
010514 000074 TCTR: .WORD 60.
010516 000000 NCOUNT: .WORD 0
010520 000000 FSTIME: .WORD 0 ;FLAG IF FIRST TIME WINDOW
010522 000000 CTRFLG: .WORD 0 ;FLAG TO SIGNAL TIME INTERRUPT ROUTINE
;TO START TIME WINDOW
010524 000000 INFLAG: .WORD 0 ;FLAG TO SIGNIFY TIME WINDOW IN PROGRESS
010526 000000 RSELECT: .WORD 0 ;FLAG TO INDICATE RUNWAY SELECTOR SWITCH
;POSITION
010530 000000 TW: .WORD 0 ;TIME WINDOW
010532 000000 TWCTR: .WORD 0 ;TIME WINDOW COUNTER,DECREMENTED BY 1
;EVERY 1/60 TH OF A SEC
010534 000000 TARSEC: .WORD 0
010536 000000 TSEC: .WORD 0

010540 000000 TMIN: .WORD 0 ;KEEPS TIME IN MINUTES
010542 000000 THR: .WORD 0 ;
010544 000000 TDAY: .WORD 0
010546 000040 MONDAY: .WORD 32.

010550 000000 TMON: .WORD 0
010552 000000 TYEAR: .WORD 0

010554 000000 ITRANS: .WORD 0 ;KEEPS TRACK OF NUMBER OF TRANS.
;WHOSE INPUTS HAVE BEEN PROCESSED FOR RVR
010556 000000 NOITER: .WORD 0 ;NO OF ITERATIONS IN RVR CALCULATIONS
010560 000000 VNEW: .WORD 0 ;TEMP STORAGE FOR RVR ITERATE
010562 000000 BALEPS: .WORD 0.0 ;STORAGE FOR D.PR. PRODUCT OF BASE*LOG10
010564 000000 ;(.05)*2**S

010566 000000 DUM: .WORD 0;TEMP
010570 000000 ITIV: .WORD 0 ;TEMP

010572 000000 CAIB: .WORD 0 ;TEMP
010574 000000 CAIK: .WORD 0 ;TEMP
010576 000000 LOGNBM: .WORD 0 ;TEMP
010600 000000 V: .WORD 0 ;TEMP
010606 VSTORE: .+.NTRDBL ;STORAGE SPACE FOR RVR FOR DIFF TRS.
010612 NPUL: .+.NTRDBL ;STORAGE SPACE FOR
;COUNTER INPUTS FROM DIFFERENT TRS
010612 000000 VINIT: .WORD 0 ;INITIAL VALUE OF RVR ITR
010614 000000 DIAGNO: .WORD 0 ;DIAGNO=1 FOR DIAGNOSTIC PRINT

010616 000011 SCALE: .WORD 9. ;SCALE FACTOR FOR INTEGER ARITHMETIC
010620 000000 MAGSCA: .WORD 0 ;=2**SCALE
010622 000000 L5280: .WORD 0 ;STORES LOG10(5280.)*2**SCALE
010624 000000 LEPS: .WORD 0 ;STORES LOG10(EPS=.05)*2**SCALE
010626 000000 MEXIT: .WORD 0 ;EXIT SCALE FACTOR

010630 000000 MODNR: .WORD 0 ;DETEMINES ITERATION METHOD TO SOLVE ALLARD'S LAW
;=0:N-R METHOD
;=1:MOD N-R METHOD

```

```

;=2:HALLEY'S METHOD
;NORMALLY MODNR=0,IF HIGHER ORDER METHOD
;USED,APPROPRIATE VALUES SHOULD BE TOGG
;-LED IN FROM
;SWITCH BOARD

```

```

;
;
;

```

```

;
;VISIBILITY PROGRAMME
;
;

```

```

010632 004567 VISIB: JSR R5,INIT ;INITIALIZE
000000

010636 004567 JSR R5,TTYPAR ;PARAMETERE INPUT FROM TTY
176466

010642 004567 VISLUP: JSR R5,INPUT ;DATA INPUT FROM TTY OR INTERFACE
176346

010646 004567 JSR R5,CALC ;DEPENDING UPON INPUT MODE
174726 ;PROCESSES DATA INPUTS TO PRODUCE

010652 004567 JSR R5,OUTPUT ;RVR VALUES
175626 ;OUTPUTS RVR AND ASSOCIATED DATA

010656 004567 JSR R5,TTYPA2 ;CHECKS ONCE EVERY NINPUT TIMES
176766

010662 000767 BR VISLUP ;IF NEW SET OF PARAMETERS ARE TO BE INPU
;BACK FOR MORE DATA INPUT
;NEED TO BE CHANGED ;IF SO IT
;LOOPS BACK TO TTYPAR BY ITSELF

010664 000000 HALT

010632 ; .END VISIB

```

```

;
;
; TTY I/O PACKAGE AND EXECUTIVE
;
; SUBROUTINES:
;   EXECUT: EXECUTIVE PROGRAMME TO MONITOR
;           THE PROGRAMME DURING RUN-TIME.
;
;
;           PROGRAMME ENTERED BY CONTROL C KEY INPUT
;           FROM TTY KB. EXECUTIVE CAN RESPOND TO A SET OF
;           COMMANDS DESCRIBED BELOW. COMMANDS ARE INPUT
;           FROM KB; AND ARE TWO CHS LONG TERMINATED
;           BY EITHER A CR OR COMMA. ANY OTHER TERMINATING
;           CHARACTER IS NOT VALID. FOR INVALID CHS. , '?' IS
;           PRINTED ON TTY AND A FRESH STRING OF COMMAND CHS
;           CAN BE INPUT.
;
;           COMMAND STRINGS:
;
;           (1) CO CONTINUE WITH THE PROGRAMME WHERE IT WAS
;               INTERRUPTED.
;           (2) AR GO TO ARITHMETIC PACKAGE TEST ROUTINE
;               ARTEST.
;           (3) IO GO TO TTY I/O PACKAGE TEST ROUTINE
;               IOTEST.
;           (4) RE RESTART VISIBILITY PROGRAMME.
;           (5) TI TIME COMMAND
;               IF TERMINATED BY CR, PRINTS TIME IN HR
;               , MIN, SEC AND CONTINUES WITH THE
;               INTERRUPTED PROGRAMME.
;
;           IF TERMINATED BY COMMA, ACCEPTS FROM KB
;           HR, MIN, SECS IN 3(I-) FORMAT DESCRIBED
;           BELOW. THEN INTERRUPTED PROGRAMME
;           IS RESUMED.
;
;           (6) DA DATE COMMAND
;               IF TERMINATED BY CR, PRINTS DATE
;               AS DAY , MONTH, YEAR AND CONTINUES
;               PROGRAMME.
;
;               IF TERMINATED BY COMMA, ACCEPTS
;               DAY, MONTH, YEAR IN (3I-) FORMAT,
;               AND CONTINUES WITH INTERRUPTED
;               PROGRAMME.
;
; IOTEST: TEST ROUTINE FOR TTY I/O PACKAGE
; INIT:   INITIALIZATION, ENABLE
;         INTERRUPTS OF PERIPHERALS, SET UP
;         STACK AREA, SET UP SERVICING ROUTINES
;
;
; FOR INTERRUPT VECTORS.
; GETCHR: GET A CHARACTER FROM TTY KB BUFFER

```

000000 000000\*

PAGE 001

```
;
; PUTCHR: PRINT A CHARACTER ON TTY
; LINE: ASSEMBLE A STRING OF CHARACTERS FORM TTY
; TERMINATED BY CR OR COMMA.EDITING PROVISION
; BY RUBOUT KEY AND DELETING OF CHARACTER STRING
; INPUT SO FAR BY CONTROL SHIFT K AND CONTROL U
; KEY INPUTS IS PROVIDED.CHARACTERS ARE APPROPRIAT
; ECHOED ON TTY.
; GETNUM: ACCEPTS A + INTEGER FROM TTY AND ASSEMBLES
; A BINARY NO.INPUT FORMAT IS I-
; WHERE I- SIGNIFIES A VARIABLE I FORMAT
; WITH INPUT TERMINATED EITHER BY A COMMA OR CR
; PROGRAMME PERSISTS TILL A VALID + INTEGER
; <2**15-1 AND WITH NO NON-NUMERIC CHS IS
; INPUT
; ICABUF: PRINTS OUTPUT LIST A1,A2,.....AN ON TTY
; FORMAT(/N(2X,I6,4X))
; ICA: AASEMBLES AN OUTPUT FORMAT(6I,4X)
; WHERE I FORMAT HAS A LEADING SIGN
; AND FIVE DIGITS CORRESPONDING TO A
; GIVEN BINARY NUMBER.
; INBUF: ACCEPTS FROM TTY INPUT LIST
; A1,A2,.....AN WITH FORMAT
; ($MES1$,I-,$MES2$,I-,...,$MESN$,I-)
; IBUFFER: ACCEPTS FROM TTY INPUT LIST
; A1,A2,.....AN IN THE FORMAT
; (NI-)
; RDRINT: SERVICES TTY KB INTERRUPT
;
;
;
```

```
000000 R0=%0
000001 R1=%1
000002 R2=%2
000003 R3=%3
000004 R4=%4
000005 R5=%5
.GLOBL GETCHR,GETNUM,ICA,ICABUF,INIT,MESS,PUTCHR,LINE,INBUF
.GLOBL VISIB,IOTEST,ARTEST,THR,TMIN,TSEC,TDAY,TMON,TYEAR
.GLOBL MONDET,MONDAY,LKSERV
```

```
000006 SP=%6
000007 PC=%7
.TITLE IOTEST
```

```
000000 .ASECT
002000 .=2000
;
```

```
;TTY I-O PACKAGE TEST PROGRAMME
;SHOULD DO THE FOLLOWING:
;1)PRINT 'VALUE=' ON TTY
;2)ACCEPT + INTEGER<2**15-1 FROM TTY KB
```

```

;3)PRINT NUMBER SO INPUT ON TTY
;

```

```

002000 004567 IOTEST: JSR R5,INIT
000046

```

```

002004 004567 MAINBAC:      JSR R5,INBUF
001140
002010 000001              .WORD 1
002012 002032              .WORD MAINMES,MAINBUF
002014 002044

```

```

002016 004567              JSR R5,ICABUF
000674
002022 000001              .WORD 1,,MAINBUF
002024 002044
002026 000766              BR MAINBAC
002030 000000              HALT
002032 015 MAINMES: .BYTE 15,12
002033 012
002034 126                  .ASCII /VALUE=/
002035 101
002036 114
002037 125
002040 105
002041 075
002042 000                  .BYTE 0
002044 000                  .EVEN

```

```

002044 000000 MAINBUF:      .WORD 0,0
002046 000000
002050 000                  .BYTE 0
002052 000                  .EVEN

```

```

;
; TTY IO PACKAGE
;-----
;
;

```

```

;SUBROUTINE INIT
;SHOULD BE INCORPORATED IN THE PROGRAMME BEFORE ANY I/O
;OPERATIONS IN THE MAIN PROGRAMME
;

```

```

002052 012706 INIT:      MOV #1000,SP      ;STACK ADDRESS
001000
002056 012737*          MOV #LKSERV,0#100      ;LINE CLOCK INTERRUPT
000000
000100
;VECTOR
002064 012737          MOV #300,0#102      ;PROCESSOR STATUS
000300

000102
002072 012737          MOV #100,0#177546      ;ENABLE LINE CLOCK
000100
177546

```



PAGE 003

```
002100 012737      MOV #RDRINT.0#60      ;READER INTERRUPT SERVICE ROUTIN
      003256
      000060
002106 012737      MOV #200.0#62        ;INTERRUPT SERVICE ROUTINE PROG
      000200
      000062
002114 012737      MOV #101.,0#177560    ;SET READER ENABLE
      000145
      177560
                                           ;AND INTERRUPT ENABLE

002122 000205      RTS R5

;
;SUBROUTINE GETS A CHARACTER FROM TTY KB
;
002124 005237  GETCHR: INC 0#177560    ;RDR ENABLE
      177560
002130 005067      CLR GETFLG        ;FLAG=0 INDICATES THAT
      001564
                                           ;CHARACTER IN BUFFER IS NOT READ
002134 005767  GEWAIT: TST GETFLG
      001560
002140 001775      BEQ GEWAIT        ;WAIT TILL BUFFER IS READ
002142 116700      MOV B RDRLOC.R0    ;RDRLOC CONTAINS CH. READ FROM BUFFER
      001550
002146 000205      RTS R5

;
;
;THIS SUBROUTINE PRINT S A STRING OF CHARACTERS POINTED
;TO BY R4 UNTILL A 0 IS ENCOUNTERED
;
002150 010046  MESS:  MOV R0,-(SP)
002152 112400  MLOOP: MOV B (R4)+,R0    ;MOV THE BYTE POINTED TO AND INCREMENT T
002154 001403      BEQ MDDONE        ;IS BYTE EQUAL TO 0?
002156 004567      JSR R5,PUTCHR
      000006

                                           ;THE REST OF THE MESSAGE
002162 000773      BR MLOOP

;
002164 012600  MDDONE: MOV (SP)+,R0
002166 000205      RTS R5

;
002170 105737  PUTCHR: TST B 0#177564    ;TELEPRINTER READY?
      177564
002174 100375      BPL PUTCHR

002176 110037      MOV B R0,0#177566    ;CHARACTER TO TELEPRINTER
      177566
002202 000205      RTS R5
;
```

```

;
;
;
; THIS SUBROUTINE GETS A LINE OF CHARACTERS FROM TTY
; AND LEAVE THEM IN THE BUFFER
;
002204 010046 LINE: MOV R0,-(SP)
002206 010146     MOV R1,-(SP)
002210 010246     MOV R2,-(SP)
002212 010346     MOV R3,-(SP)
002214 010446     MOV R4,-(SP)
;
002216 012701 LENTER: MOV#BUFFER,R1 ;BUFFER POINTER
002410
002222 012702     MOV #BUFEND,R2 ;BUFFER END POINTER
002520
002226 010103     MOV R1,R3 ;BEGINNING OF BUFFER
002230 005004     CLR R4 ;RUBOUT FLAG
;
002232 020102 LLOOP: CMP R1,R2 ;AT THE END?
002234 001450     BEQ TOOBIG
002236 004567     JSR R5,GETCHR ;GET CHARACTER FROM TTY
002242 120027     CMPB R0,#177 ;RUBOUT?
000177
002246 001525     BEQ RUBOUT
002250 120027     CMPB R0,#33 ;ESCAPE(CONTROL SHIFT K KEY)
000033
002254 001450     BEQ DELETE
002256 120027     CMPB R0,#25 ;CONTROL U KEY?
000025
002262 001445     BEQ DELETE
002264 005704     TST R4 ;WAS RUBOUT PREVIOUS INPUT?
002266 001407     BEQ LOK ;IF NOT BRANCH
002270 010046     MOV R0,-(SP) ;IF SO SAVE CH FROM KB
002272 012700     MOV #57,R0 ;FILE SEPERATOR CHARACTER
000057
002276 004567     JSR R5,PUTCHR
002302 005004     CLR R4
002304 012600     MOV(SP)+,R0 ;RESTORE CH FORM STACK
;
002306 110021 LOK: MOVB R0,(R1)+ ;LOAD INTO BUFFER AND INCREMENT POINTER
002310 120027     CMPB R0,#15 ;CARRIAGE RETURN?
000015
002314 001413     BEQ LDONE1 ;IF SO DONE
002316 004567     JSR R5,PUTCHR
002322 120027     CMPB R0,#54 ;COMMA?
000054
002326 001341     BNE LLOOP ;IF NOT BACK TO GET MORE CHARACTERS FROM
;
002330 012604 LDONE2:MOV(SP)+,R4 ;RESTORE REG. CONTENTS FROM STACK
;BEFORE EXIT
002332 012603     MOV (SP)+,R3
002334 012602     MOV (SP)+,R2

```

PAGE 005

```
002336 012601      MOV(SP)+,R1
002340 012600      MOV (SP)+,R0
002342 000205      RTS R5
;
002344 012704 LDONE1: MOV #CRLF,R4      ;CR AND LF CHS FOR TELEPRINTER
002405
002350 004567      JSR R5,MESS
002354 000765      BR LDONE2      ;TO EXIT
;
002356 012704 T00BIG: MOV#LBIG,R4      ;REENTER ROUTINE TO SET UP BUFFER AGAIN
002370
;
002362 004567 TWOBIG: JSR R5,MESS
002366 000713      BR LENTER
;
002370 015 LBIG: .BYTE 15,12
002371 012
002372 077 .ASCII /?/
002373 015 .BYTE 15,12,0
002374 012
002375 000
;
002376 012704 DELETE: MOV #LDEL,R4      ;PRINT MESSAGE 'D'AND BRANCH
002404
002402 000767      BR TWOBIG
;
002404 104 LDEL: .ASCII /D/
002405 015 CRLF:.BYTE 15,12,0
002406 012
002407 000
002410 .EVEN
002520 BUFFER: .+.110 ;BUFFER AREA
002522 BUFEND: .+.2
002522 020103 RUBOUT: CMP R1,R3      ;POINTER AT BEG. OF BUFFER
002524 001642      BEQ LLOOP
;
;LOOP TO GET OTHER CH.
002526 005704      TST R4      ;CH IF THERE WAS A PREVIOUS RUBOUT
002530 001005      BNE DIRTY
002532 012700      MOV #57,R0
002536 004567      JSR R5,PUTCHR      ;PUNCH IT AND RESET FLAG
002542 005204      INC R4
002544 114100 DIRTY: MOVB-(R1),R0      ;DEC POINTER,UNLOAD CH IN BUFFER
002546 004567      JSR R5,PUTCHR
002552 000627      BR LLOOP      ;BACK TO LOOP
;
;
;ROUTINE TO ASSEMBLE A NUMERIC VALUE FROM TTY

;RETURNS WITH POSITIVE NUMBER IN BINARY IN R0
;ERROR MESSAGE "RETYPE" ON TTY FOR THE FOLLOWING:
;(1)A NON NUMERIC CH INCLUDING SIGN CHS (2) NUMBER TYPED
;IN IS LARGER THAN 2^15-1. AFTER THE ERROR MESSAGE
```

```

;RETYPE THE NUMBER
;
;
002554 010146 GETNUM: MOV R1,-(SP)
002556 010246         MOV R2,-(SP)
002560 010446         MOV R4,-(SP)
002562 012701 GBACK:MOV#BUFFER,R1
          002410
002566 005002         CLR R2
002570 004567         JSR R5,LINE
          177410
002574 112100 GLOOP1:MOV B (R1)+,R0
002576 120027         CMPB R0,#15      ;CR?
          000015
002602 001425         BEQ GDONE1
002604 120027         CMPB R0,#54      ;COMMA?
          000054
002610 001422         BEQ GDONE1
002612 162700         SUB #60,R0
          000060
002616 100424         BMI GBAD        ;TO BINARY
002620 022700         CMP #12,R0
          000012
002624 003421         BLE GBAD        ;IF NOT
002626 006302         ASL R2
002630 102417         BVS GBAD
002632 010204         MOV R2,R4
002634 006302         ASL R2
002636 102414         BVS GBAD
002640 006302         ASL R2
002642 102412         BVS GBAD
002644 060402         ADD R4,R2      ;R2=R2*10
002646 102410         BVS GBAD
002650 060002         ADD R0,R2      ;R2=R2*10 +R0
002652 102406         BVS GBAD      ;OVERFLOW I.E. MAG OF NO IS GREATER THAN
002654 000747         BR GLOOP1      ;
002656 010200 GDONE1: MOV R2,R0
002660 012604         MOV (SP)+,R4    ;RESTORE
002662 012602         MOV (SP)+,R2    ;R4,R2,R1
002664 012601         MOV (SP)+,R1
002666 000205         RTS R5
002670 012704 GBAD:  MOV #GEVIL,R4
          002702
002674 004567         JSR R5,MESS
          177250
002700 000730         BR GBACK
002702 015         GEVIL: .BYTE 15,12 ;CRLF
002703 012
002704 122         .ASCII /RETYPE/
002705 105
002706 124
002707 131

002710 120
002711 105
002712 015         .BYTE 15,12,0
002713 012

```

```

002714 000
002716 .EVEN

;OUTPUTS DATA ON TTY
;CALLING SEQUENCE:
; JSR R5,ICABUF
; .WORD N ;NO. OF VARIABLES IN LIST
; .WORD A1,A2,.....,AN
; ;OUPUT LIST OF VARIABLES
;
;FORMAT OF OUTPUT:(/N(2X,I6.4X))
;
;
;
002716 010046 ICABUF: MOV R0,-(SP)
002720 010246 MOV R2,-(SP)
002722 010446 MOV R4,-(SP)
002724 012702 MOV#BUFFER,R2
002410
002730 012567 MOV (R5)+,WORCNT ;NO OF WORDS
000046
002734 013500 ICALUP: MOV@(R5)+,R0 ;R0
002736 004567 JSR R5,ICA ;CONVERTS BINARY TO ASCII,LOADS SIGN
000042
; PLUS FIVE DIGITS PLUS FOUR
; SPACES INTO BUFFER
002742 005367 DEC WORCNT
000034
002746 003372 BGT ICALUP ;LOOP BACK IF ALL WORDS NOT DONE
002750 112722 MOVB #15,(R2)+
000015
002754 112722 MOVB #12,(R2)+
000012
002760 105012 CLRB (R2) ;IF DONE LOAD CRLF AND 0
002762 012704 MOV #BUFFER,R4
002410
002766 004567 JSR R5,MESS ;PRINT THE NUMBER IN THE BUFFER
177156
002772 012604 MOV (SP)+,R4
002774 012602 MOV (SP)+,R2
002776 012600 MOV (SP)+,R0
003000 000205 RTS R5
;
003002 000000 WORCNT: .WORD 0
;
;
;
;ICA SUBROUTINE
;CONVERTS BINARY INTEGER IN R0 TO ASCII CODE
;AT ADDRESS BUFFER, BUFFER CONTAINS LEADING SIGN BYTE
;FOLLOWED BY FIVE NUMERICAL DIGITS AND FOUR SPACE

;ASCII BYTES, BUFFER ADDREDD AVAILABLE IN R2
;
003004 010146 ICA: MOV R1,-(SP)
003006 010346 MOV R3,-(SP)

```

PAGE 010

```
003010 010446      MOV R4,-(SP)
003012 012701      MOV #DWORD,R1
          003134
003016 005700      TST R0 ;CHECK SIGN
003020 100403      BMI ICA3
003022 112722      MOVB #53,(R2)+ ;LOAD ASCII CH. FOR +
          000053
003026 000404      BR ICLOOP
003030 112722 ICA3: MOVB #55,(R2)+ ;-ASCII CH
          000055
003034 005400      NEG R0
003036 102427      BVS ICAERR ;NEGMAX
003040 005003 ICLOOP: CLR R3 ;SCALE FLAG
003042 011104      MOV(R1),R4 ;ELEMENT OF DECIMAL ARRAY
003044 001414      BEQ ICEND ;IF EQ 0,DONE
003046 020004 ICA2: CMP R0,R4 ;R0<R4?
003050 002404      BLT ICA1 ; YES.GO TO NEXT ELEMENT
003052 005203      INC R3 ;INCREMENT FACTOR
003054 061104      ADD (R1),R4 ;NO.ADD ARRAY ELEMENT TO R4
003056 102401      BVS ICA1 ;OVERFLOW
003060 000772      BR ICA2 ;LOOP BACK
;
003062 062703 ICA1: ADD #60,R3 ;CONVERT SCALE TO ASCII
          000060
003066 110322      MOVB R3,(R2)+ ;LOAD SCALE ONTO BUFFER
003070 160400      SUB R4,R0 ;FORM REMAINDER
003072 062100      ADD (R1)+,R0 ;IN R0
003074 000761      BR ICLOOP
003076 012703 ICEND: MOV #020040,R3 ;TWO ASCII SPACE CHS.
          020040
003102 010322      MOV R3,(R2)+
003104 010322      MOV R3,(R2)+
003106 012604      MOV (SP)+,R4
003110 012603      MOV (SP)+,R3
003112 012601      MOV (SP)+,R1
003114 000205      RTS R5
003116 112722 ICAERR: MOVB #63,(R2)+
          000063
003122 012722      MOV #031067,(R2)+
          031067
003126 012722      MOV #033070,(R2)+ ;LOAD -32768 IN BUFFER
          033070
003132 000761      BR ICEND
003134 023420 DWORD: .WORD 10000,,1000,,100,,10,,1,,0
003136 001750
003140 000144
003142 000012
003144 000001
003146 000000
;
;SUBROUTINE TO HANDLE INPUT FROM TTY
;CALLING SEQUENCE
;
; JSR R5,INBUF
; .WORD N ;NO.OF VARIABLES IN INPUT LIST
```

```

;
;           ;N PAIRS OF WORDS FOLLOW
;           ;FIRST WORD OF PAIR CONTAINS ADD
;           ;VARIABLE MESSAGE TO TTY
;           ;SECOND WORD THE VARIABLE ADDRESS
;
;           .WORD AMESS,A
;           .WORD BMESS,B
;
;           .WORD NMESS,N
;
;           ;PROGRAMME RETURNS HERE
;

003150 010046 INBUF:  MOV R0,-(SP)
003152 010146          MOV R1,-(SP)
003154 010216          MOV R2,-(SP)
003156 010346          MOV R3,-(SP)
003160 010446          MOV R4,-(SP)
003162 012503          MOV (R5)+,R3      ;COUNT OF VARIABLES TO BE INPUT
003164 012700 INBU1:  MOV #020040,R0 ;TWO ASCII SPACE CHS.
                   020040
003170 004567          JSR R5,PUTCHR   ;TWO BLANK SPACES ONTO TTY
                   176774
003174 012504          MOV (R5)+,R4      ;POINTER TO THE MESSAGE TO TTY
003176 004567          JSR R5,MESS     ;PRINT THE MESSAGE ON TTY
                   176746
003202 004567          JSR R5,GETNUM   ;GET THE INPUT FROM TTY
                   177346
003206 010035          MOV R0,@(R5)+   ;MOV THE INPUT TO THE VARIABLE ADDRESS
003210 005303          DEC R3         ;DECREMENT THE COUNT
003212 001364          BNE INBU1      ;BACK, IF ALL VARIABLES NOT INPUT
;
003214 012604          MOV (SP)+,R4
003216 012603          MOV (SP)+,R3
003220 012602          MOV (SP)+,R2
003222 012601          MOV (SP)+,R1
003224 012600          MOV (SP)+,R0
003226 000205          RTS R5
;
;SUBROUTINE TO HANDLE INPUT FROM TTY
;FORMAT (NI-),- FOLLOWING I INDICATES
;THAT WORD IS INPUT IN I FORMAT AND COULD BE TERMINATED
;BY A CR OR COMMA
;
003230 010046 IBUF:   MOV R0,-(SP)
003232 010446          MOV R4,-(SP)
003234 012504          MOV (R5)+,R4      ;NO OF WORDS TO BE INPUT
003236 004567 IUFBA: JSR R5,GETNUM
                   177312
003242 010035          MOV R0,@(R5)+
003244 005304          DEC R4
003246 001373          BNE IUFBA
003250 012604          MOV (SP)+,R4

003252 012600          MOV (SP)+,R0
003254 000205          RTS R5
;

```

```

;
;
;TTY INTERRUPT SERVICING ROUTINE
;
003256 113767 RDRINT: MOVB @#177562,RDRLOC ;MOVE BYTE
177562
000432
003264 042767 BIC #177600,RDRLOC ;ASCII CODE
177600
000424
003272 126727 CMPB RDRLOC,#3 ;CONTROL C KEY INPUT?
000420
000003
003300 001403 BEQ RDR1 ;IF SO BRANCH
003302 005267 INC GETFLG
000412

003306 000002 RTI ;IF NOT RETURN TO THE POINT OF
;INTERRUPTION
- 003310 016646 RDR1: MOV 2(SP),-(SP) ;SET UP PROGRAMME STATUS FOR
000002
;EXECUT
003314 012746 MOV #EXECUT, -(SP)
003322
003320 000002 RTI
;
;
;EXECUTIVE PROGRAMME
;ENTERED FROM TTY KB BY CONTROL C KEY INPUT.
;A VARIETY OF EXECUTIVE OR MONITOR FUNCTIONS COULD BE INCLUDED H
;
003322 016746 EXECUT: MOV BUFFER, -(SP)
177062
003326 016746 MOV BUFFER+2, -(SP) ;SAVE PREVIOUS CONTENTS OF BUFFE
177060
;ONLY FIRST TWO WORDS SAVED
003332 010046 MOV R0, -(SP)
003334 010146 MOV R1, -(SP)
003336 010246 MOV R2, -(SP)
003340 010346 MOV R3, -(SP)

003342 010446 MOV R4, -(SP)
003344 010546 MOV R5, -(SP)

003346 112700 MOVB #'..R0
000056
003352 004567 JSR R5,PUTCHR ;PRINT ',' ON TTY
176612
003356 004567 EXELUP: JSR R5,LINE ;GET A LINE OF CHS. FROM KB
176622

003362 016700 MOV BUFFER,R0 ;ADDRESS OF FIRST WORD OF INPUT CHS
177022

003366 020027 CMP R0,#"CO ;R0 CONTAINS FIRST TWO BYTES
047503 ;=CO?

```



PAGE 013

```
003372 001424      BEQ EXECDN
003374 020027      CMP R0,#"RE      ;=RE?
                042522
003400 001436      BEQ EXERES
003402 020027      CMP R0,#"AR      ;=AR?
                051101
003406 001435      BEQ EXEART
003410 020027      CMP R0,#"IO      ;=IO?
                047511
003414 001434      BEQ EXEIOT
003416 020027      CMP R0,#"TI      ;=TI?
                044524
003422 001433      BEQ EXETIM
003424 020027      CMP R0,#"DA      ;=DA?
                040504
003430 001472      BEQ EXEDAY
003432 112700 EXEOUT: MOV B #'?.R0
                000077
003436 004567      JSR R5,PUTCHR   ;IF NONE OF ABOVE PRINT '?' ON TTY
                176526
003442 000745      BR EXELUP      ;AND BACK TO GET CORRECT COMMAND
                ;
003444 012605 EXECON: MOV (SP)+,R5
003446 012604      MOV (SP)+,R4
003450 012603      MOV (SP)+,R3
003452 012602      MOV (SP)+,R2
003454 012601      MOV (SP)+,R1

003456 012600      MOV (SP)+,R0
003460 012667      MOV (SP)+,BUFFER+2
                176726
003464 012667      MOV (SP)+,BUFFER      ;RESTORE BUFFER CONTENTS
                176720
003470 005067      CLR GETFLG      ;SET THE GETCHR ROUTINE FLAG=0
                000224

003474 000002      RTI
                ;
003476 004567 EXERES: JSR R5,VISIB
                000000
                ;
003502 004567 EXEART: JSR R5,ARTEST
                000000
                ;
003506 004567 EXEIOT: JSR R5,IOTEST
                176266
                ;
003512 126727 EXETIM: CMPB BUFFER+2,#15      ;CR?
                176674
                000015
003520 001427      BEQ EXETI1

003522 126727      CMPB BUFFER+2,#54      ;COMMA?
                176664
                000054
003530 001340      BNE EXEOUT      ;IF 'TI' IS NOT TERMINATED BY CR OR
```

PAGE 014

```
                                ;COMMA BACK TO GET CORRECT COMMAND
003532 004567      JSR R5,IBUFER
                                177472
003536 000003      .WORD 3
003540 000000*    .WORD THR,TMIN,TSEC
003542 000000*
003544 000000*
003546 026727*    CMP THR,#23.      ;THR>23?
                                000000
                                000027
003554 003326      BGT EXEOUT
003556 026727*    CMP TMIN,#59.    ;TMIN>59?
                                000000
                                000073
003564 003322      BGT EXEOUT
003566 026727*    CMP TSEC,#59.   ;SEC>59?
                                000000
                                000073
003574 003316      BGT EXEOUT
                                ;IF TIME INPUT IS INVALID
                                ;'?' IS PRINTED ON TTY
                                ;INPUT COMMAND STRING AGAIN

003576 000722      BR EXECON

                                ;
003600 004567 EXETI1: JSR R5,ICABUF
                                177112
003604 000003      .WORD 3
003606 000000*    .WORD THR,TMIN,TSEC
003610 000000*
003612 000000*

003614 000713      BR EXECON

                                ;
003616 126727 EXEDAY: CMPB BUFFER+2,#15      ;CR?
                                176570
                                000015
003624 001425      BEQ EXEDA1
003626 126727      CMPB BUFFER+2,#54      ;COMMA?
                                176560
                                000054
003634 001276      BNE EXEOUT

003636 004567      JSR R5,IBUFER      ;ACCEPT DAY,MONTH AND YEAR FROM KB
                                177366
003642 000003      .WORD 3
003644 000000*    .WORD TDAY,TMON,TYEAR
003646 000000*
003650 000000*
003652 026727*    CMP TMON,#12.    ;TMON>12?

                                000000
                                000014
003660 003264      BGT EXEOUT
```

PAGE 015

```
003662 004567 JSR R5,MONDET
000034
003666 026767* CMP TDAY,MONDAY ;TDAY,GE.MAXIMUM DAYS
000000
000000
003674 002256 BGE EXECUT ;INA MONTH+1

;IF DATE INPUT IS ILLEGAL
;'?' IS PRINTED ON TTY
;AND FRESH COMMAND STRING CAN BE INPUT

003676 000662 BR EXECON ;CONTINUE WITH THE PROGRAMME
003700 004567 EXEDA1: JSR R5,ICABUF
177012
003704 000003 .WORD 3
003706 000000* .WORD TDAY,TMON,TYEAR ;PRINT DAY,MON,YEAR ON TTY
003710 000000*
003712 000000*
003714 000653 BR EXECON
;
003716 000000 RDRLOC: .WORD 0
003720 000000 GETFLG: .WORD 0

;FOLLOWING ROUTINE DETERMINES DAYS IN A MONTH
;LEAP YEAR ALSO ACCOUNTED FOR
;MONDAY=MAXIMUM NO OF DAYS IN THE MONTH+1
;
003722 010046 MONDET:MOV R0,-(SP)
003724 016700* MOV TMON,R0
000000
003730 020027 CMP R0,#7 ;COMPARE TO JULY?
000007
003734 003016 BGT EXE2 ;BRANCH IF GREATER
003736 020027 CMP R0,#2 ;FEB?
000002

003742 001416 BEQ EXE3
003744 006200 EXE7: ASR R0 ;DIVIDE BY 2
003746 103404 BCS EXE4 ;JUMP IF ODD MONTH
003750 012767* MOV #31,,MONDAY ;=30+1
000037
000000
003756 000403 BR EXE5
003760 012767* EXE4: MOV #32,,MONDAY ;=31+1
000040
000000
003766 012600 EXE5: MOV (SP)+,R0
003770 000205 RTS R5

;
003772 162700 EXE2: SUB#7,R0 ;FOR MONTHS STARTING WITH AUGUST
000007
003776 000762 BR EXE7
```

```
                ;FOR FEBRUARY
004000 016700*EXE3:  MOV TYEAR,R0
                000000
004004 006200      ASR R0 ;DIVIDE BY 2
004006 103406      BCS EXE6 ;CANNOT DIVIDE BY 2
004010 006200      ASR R0 ;FURTHER DIVIDE BY 2
004012 103404      BCS EXE6 ;CANNOT DIVIDE
004014 012767*    MOV #30.,MONDAY ;=29+1
                000036
                000000
004022 000761      BR EXE5
004024 012767*EXE6:  MOV #29.,MONDAY ;=28+1
                000035
                000000
004032 000755      BR EXE5
                ;
                ;
                ;
002000 .END IOTEST
```

000000 ERRORS

```

;
;ARITHMETIC PACKAGE
;
;SUBROUTINES
;   ARTEST: ARTH.PACKAGE SOFTWARE TEST ROUTINE
;   MULT:   MULTIPLY ROUTINE
;   DIV:    DIVIDE ROUTINE
;   FMULT:  SPECIAL MULTIPLY ROUTINE
;   FDIV:   SPECIAL DIVIDE ROUTINE
;   LOG:    LOGARITHM ROUTINE, BASE 10.
;   NORMD:  DOUBLE PRECISION NORMALIZATION ROUTINE
;   SALOG:  SUPPLIES SCALED LOGARITHM.
;   RSHIFT: RIGHT SHIFT ROUTINE
;   LSHIFT: LEFT SHIFT ROUTINE
;   NORM:   NORMALIZATION ROUTINE
;
;

```

```

000000 R0=%0
000001 R1=%1
000002 R2=%2
000003 R3=%3
000004 R4=%4
000005 R5=%5
000006 SP=%6
000007 PC=%7

```

```

.TITLE ARTEST
000000 .ASECT
.GLOBL INBUF, MESS, ICABUF, INIT, GETNUM, ARTEST

```

```

004400 . =4400
.GLOBL LSHIFT, SCALE, RSHIFT, DIV, MULT, LOG, SCALOG, NORM, FMULT, FDIV

```

```

;MULTIPLY ROUTINE
; SUPPLIES DOUBLE PRECISION INTEGER PRODUCT IN (R0,R1)
; WITH SIGNIFICANT PRODUCT IN R1 AND OVERFLOW IN R0.
;MULTIPLICAND AND MULTIPLIER IN R0 AND R2
;
;
;

```

```

004400 005001 MULT: CLR R1 ;COUNTER
004402 010446     MOV R4, -(SP)
004404 005004     CLR R4 ;SIGN
004406 005700     TST R0 ;MULTIPLICAND
004410 001437     BEQ MULZER ;ZERO PRODUCT
004412 100003     BPL MULT1 ;JUMP IF +
004414 005204     INC R4 ;NOTE -
004416 005400     NEG R0
004420 102436     BVS MULERR
;

```

```

004422 005702 MULT1: TST R2
004424 001431     BEQ MULZER

```

PAGE 001

```
004426 100003      BPL MULT2      ;JUMP IF -
004430 005204      INC R4      ; SET PRODUCT SIGN
004432 005402      NEG R2
004434 102430      BVS MULERR
;
004436 010346      MULT2: MOV R3,-(SP)
004440 012703      MOV #-16.,R3      ;SET CO"NTER
177760
;
004444 006301      MLOOP: ASL R1
004446 006100      ROL R0      ;DOUBLE PRECISION LEFT SHIFT
004450 103002      BCC NOADD      ; MOST SIG BIT GOVERNS ADD
004452 060201      ADD R2,R1      ;IF SET ADD MULT.
004454 005500      ADC R0      ;KEEP 32 BIT PRODUCT
;
004456 005203      NOADD:INC R3      ;DONE?
004460 001371      BNE MLOOP      ; IF NOT CONTINUE
004462 012603      MOV (SP)+,R3
004464 006004      ROR R4      ;GET PRODUCT SIGN
004466 103401      BCS MULOUT      ;JUMP IF -
004470 000405      BR MULT3
;
004472 005100      MULOUT: COM R0
004474 005101      COM R1
004476 062701      ADD #1,R1
000001
004502 005500      ADC R0
004504 012604      MULT3: MOV (SP)+,R4
004506 000205      RTS R5
;
;
004510 005000      MULZER: CLR R0      ;ZERO PRODUCT
004512 005001      CLR R1
004514 000773      BR MULT3
;
004516 012704      MULERR: MOV#MULMES,R4
004530
004522 004567      JSR R5,MESS
000000
004526 000766      BR MULT3
;
004530 015      MULMES: .BYTE 15,12
004531 012
004532 116      .ASCII /NEGMAX,MUL/
004533 105
004534 107
004535 115
004536 101.
004537 130
004540 054
004541 115

004542 125
004543 114
004544 015      .BYTE 15,12,0
004545 012
```

```

004546 000
004550 .EVEN
;
;
;DIVIDE ROUTINE
;DOUBLE PRECISION DIVIDEND REQUIRED
;(R0,R1) CONTAINS DOUBLE PRECISION DIVIDED
;R2 DIVIDER
;QUOTIENT IN R1
;
;
004550 010346 DIV: MOV R3,-(SP)
004552 005003 CLR R3 ;SIGN
004554 005702 TST R2 ;DENOMINATOR
004556 001447 BEQ DIVERR ;CANNOT DIVIDE BY ZERO
004560 100002 BPL DIV1 ;JUMP IF +
004562 005203 INC R3 ;NOTE-
004564 005402 NEG R2
;
004566 005700 DIV1: TST R0 ;CHECK NUMERATOR
004570 001436 BEQ DIV5
004572 100006 BPL DIV2 ;JUMP IF +
004574 005203 INC R3 ;SET RESULT SIGN
004576 005100 COM R0
004600 005101 COM R1
004602 062701 ADD #1,R1
000001
004606 005500 ADC R0
;
004610 020200 DIV2: CMP R2,R0
004612 103437 BLO DIVER2
004614 010446 MOV R4,-(SP)
;
004616 012704 MOV #16.,R4 ;SET FOR 16 ITERATIONS
000020
;
004622 006301 DIV3: ASL R1
004624 006100 ROL R0 ;DOUBLE PRECISION SHIFT
004626 001405 BEQ DIVLUP ;JUMP IF R0=0
004630 005201 INC R1 ;ASSUME IT WILLGO.INSERT QUOTINT BIT
004632 160200 SUB R2,R0 ;TRIAL STEP
004634 103002 BHS DIVLUP ;OK
004636 060200 ADD R2,R0 ;DIVIDEND NOT BIG ENOUGH YET
004640 005301 DEC R1 ;TAKE OUT QUOTIENT BIT
004642 005304 DIVLUP:DEC R4
004644 003366 BGT DIV3 ;GO AGAIN
004646 012604 DIV6: MOV (SP)+,R4
004650 005401 NEG R1 ;TEST FOR NEGMAX
004652 006203 ASR R3 ;GET RESULT SIGN
004654 103402 BCS DIV4 ;JUMP IF NEG
004656 005401 NEG R1 ;ANSWER IS +
004650 102406 BVS DIVERR ; JUMP IF ANSWER IS NEGMAX
;
004662 012603 DIV4: MOV(SP)+,R3
004664 000205 RTS R5 ;QUOTIENT IN R1
;
004666 005701 DIV5: TST R1

```

PAGE 003

```
004670 001347      BNE DIV2
004672 005001 DIVZER: CLR R1 ; RESULT IS ZERO
004674 000772      BR DIV4
;
004676 010446 DIVERR: MOV R4,-(SP)
004700 012704      MOV #DIVMES,R4
004734
004704 004567*      JSR R5,MESS
000000
004710 012604      MOV (SP)+,R4

004712 010446 DIVER2: MOV R4,-(SP)
004714 012704      MOV #DIMES2,R4
004747
004720 004567*      JSR R5,MESS
000000
004724 012701      MOV #070000,R1 ;SET MAG OF QUOTIENT=28672.
000000
004730 000746      BR DIV6
004732 000753      BR DIV4
;
;
004734 015 DIVMES: .BYTE 15,12
004735 012
004736 104 .ASCII /DIV BY 0./
004737 111
004740 126
004741 040
004742 102
004743 131
004744 040
004745 060
004746 054
004747 105 DIMES2: .ASCII /ERR,DIV/
004750 122
004751 122
004752 054
004753 104
004754 111
004755 126
004756 015 .BYTE 15,12,0
004757 012
004760 000
004762 .EVEN
;
;
;DIVIDE ROUTINE --FIXED POINT ARITHMETIC
;NUMERATOR IN R0
;DENOMINATOR IN R2
; QUOTIENT IN R1
;
;

004762 005001 FDIV: CLR R1
004764 006200      ASR R0
004766 006001      ROR R1 ;SET UP DIVIDEND FOR INTEGER DIV ROUTINE
004770 004567      JSR R5,DIV ;CALL INT\GER DIVIDE ROUTINE
177554
```



```

004774 000205      RTS R5
;
;
;MULTIPLY ROUTINE--FIXED POINT ARITHMETIC
;MULTIPLICAND IN R0
;MULTIPLIER IN R2
;
;
004776 004567 FMULT: JSR R5,MULT
                177376
005002 006301      ASL R1
005004 006100      ROL R0
005006 000205      RTS R5
;
;
;
;
;LOGARITHM SUBROUTINE
;SUPPLIES LOGARITHM TO BASE 10 OF INTEGER NUMBER
;INTEGER (I) AVAILABLE IN R0
;PROGRAMME RETURNS WITH LOG10(I) IN
;(R0,R1) WITH SIGNIFICANT PART IN R0
;ASSOCIATED SCALE FACTOR IN R2
;
;
;CONSTANTS
                026501      CL0=11585.
                110322      CL1=-28462.
                155364      CL3=-9484.
                164354      CL5=-5908.
                154570      L102=-9864.
;
005010 010346 LOG:  MOV R3,-(SP)
005012 010446      MOV R4,-(SP)
005014 005700      TST R0
005016 003502      BLE LOGERR      ;BRANCH IF LESS THAN OR EQUAL TO ZERO
005020 012701      MOV #15.,R1      ;SCALE FACTOR OF ARG I
                000017
005024 004567      JSR R5,NORM      ;NORMALIZE ARG
                000426
005030 010146      MOV R1,-(SP)      ;PUSH SCALE AT 15
005032 010046      MOV R0,-(SP)      ;STORE ARG AT 0
005034 006200      ASR R0      ;ARG AT 1
005036 062700      ADD #CL0,R0      ;ARG+1/SQRT2 AT 1
                026501
005042 010002      MOV R0,R2
005044 162600      SUB (SP)+,R0      ;-(ARG-1/SQRT2) AT 1
005046 004567      JSR R5,FDIV
                177710
005052 010146      MOV R1,+(SP)      ;Z AT 0

005054 010100      MOV R1,R0
005056 010102      MOV R1,R2
005060 004567      JSR R5,FMULT      ;Z2 AT J
                177712

```

PAGE 005

```
005064 010046      MOV R0,-(SP)      ;STORE Z2
005066 012702      MOV #CL5,R2
          164354
005072 004567      JSR R5,FMULT      ;C5*Z2 AT 0
          177700
005076 062700      ADD #CL3,R0       ;C3+C5*Z2 AT 0
          155364
005102 012602      MOV (SP)+,R2     ;Z2 AT 0
005104 004567      JSR R5,FMULT      ;R0=Z2*(C3+C5*Z2) AT 0
          177666
005110 062700      ADD #CL1,R0       ;R0=C1+Z2*(C3+C5*Z2)AT 0
          110322
005114 012602      MOV (SP)+,R2     ;Z AT 0
005116 004567      JSR R5,FMULT      ;R0=Z*(C1+Z2*(C3+C5*Z2))
          177654
005122 012701      MOV #L102,R1     ;-L10(2) AT 0
          154570
005126 006201      ASR R1            ;DIVIDE BY 2
005130 060001      ADD R0,R1        ;R1=-LOG10(2)/2+Z*(C1+Z2*(C3+C5*Z2))
          ;
          ;R1=LOG10(FACTION)
005132 012600      MOV (SP)+,R0     ;SCALE AT 15
005134 010146      MOV R1,-(SP)     ;L10(FR) AT 0
005136 000300      SWAB R0          ;EIGHT LEFT SHIFTS
005140 006300      ASL R0
005142 006300      ASL R0
005144 006300      ASL R0           ;THREE MORE SHIFTS, ELEVEN IN ALL
          ;SCALE AT 4
005146 012702      MOV #L102,R2     ;-L10(2) AT 0
          154570
005152 005402      NEG R2           ;L10(2) AT 0
005154 004567      JSR R5,FMULT
          177616
005160 005003      CLR R3          ;SET UP LOG10(FR) AT0 IN (R2,R3) IN DBL.
005162 012602      MOV (SP)+,R2
005164 012704      MOV #4,R4       ;COUNTER FOR SHIFTS
          000004
005170 006202 LOG1: ASR R2      ;LOW ORDER BIT TO C
005172 006003      ROR R3         ;C TO HIGH ORDER BIT OF R3
005174 005304      DEC R4
005176 001374      BNE LOG1       ;CONTINUE, IF NOT ZERO
          ;(R2,R3) CONTAINS LOG10(FR) AT 0
005200 060301      ADD R3,R1      ;L10(I)=L10(FR)+L10(EXPONENT) AT $
005202 005500      ADC R0        ;L10(I) IS IN (R0,R1) AT $
005204 060200      ADD R2,R0
005206 012702      MOV #4,R2     ;SCALE=4
          000004
005212 004567      JSR R5,NORMD   ;NORMALIZATION
          000052
          ;L10(I) IN(R0,R1) WITH SCALE FACTOR (R2)

005216 012604 LOG2: MOV (SP)+,R4
005220 012603      MOV (SP)+,R3
005222 000205      RTS R5
```

```

;
005224 005000 LOGERR: CLR R0
005226 005001 CLR R1
005230 012704 MOV #LOGMES,R4
005242
005234 004567 JSR R5,MESS
000000
005240 000766 BR LOG2
005242 015 LOGMES: .BYTE 15.12
005243 012
005244 132 .ASCII /ZERO RETURNED,LOG/
005245 105
005246 122
005247 117
005250 040
005251 122
005252 105
005253 124
005254 125
005255 122
005256 116
005257 105
005260 104
005261 054
005262 114
005263 117
005264 107
005265 015 .BYTE 15.12.0
005266 012
005267 000
005270 .EVEN
;
;
;DOUBLE PRECISION NORMALIZATION ROUTINE
;FOR LOG ROUTINE ONLY
;(R0,R1) CONTAINS DOUBLE PRECISION NUMBER
;(R2) CONTAINS SCALE OF THE NUMBER
;
005270 005302 NORMD: DEC R2 ;SCALE
005272 006301 ASL R1
005274 006100 ROL R0 ;DOUBLE PRECISION SHIFT
005276 001405 BEQ NORFIN ;OPERATION COMPLETE IF R0 IS 0
005300 102373 BVC NORMD ;CONTINUE IF SIGN DID NOT CHANGE
005302 006000 ROR R0 ;RESTORE SIGN
005304 006001 ROR R1 ;AND THE NUMBER
005306 005202 INC R2 ;AND THE SCALE
005310 000406 BR NORXIT ;NORMALIZATION COMPLETE
005312 006000 NORFIN: ROR R0 ;RESTORE SIGN:000000 OR 100000
005314 006001 ROR R1
005316 006200 ASR R0
005320 006001 ROR R1 ;AND REPLICATE IT:000000 OR 100000
005322 005202 INC R2

005324 005202 INC R2 ;AND RESTORE THE SCALE
005326 000205 NORXIT: RTS R5

```

```

;
; SCALING ROUTINE FOR LOGARITHM ROUTINE
; R0 CONTAINS INTEGER I
; (R0,R1) ON RETURN FROM LOG ROUTINE CONTAINS LOG(I)*2**11
; SCALING NEEDED IS LOG(I)*2**SCALE
; ASSUMES SCALE<11
; LOG(I)*2**SCALE AVAILABLE IN R0
;
005330 010146 SCALOG: MOV R1,-(SP)
005332 010246          MOV R2,-(SP)
005334 004567          JSR R5,LOG          ;LOG ROUTINE
          177450
005340 012701          MOV#15.,R1
          000017

005344 160201          SUB R2,R1          ;SCALE OF NO IN (R0,R1) FROM LOG
005346 166701          SUB SCALE,R1          ;
          000000
005352 001403          BEQ SCAXIT
005354 100405          BMI SCALO1

;WE NEED (R2-SCALE) RIGHT SHIFTS ON R0

005356 004567          JSR R5,RSHIFT        ;PERFORMS (R1) RIGHT SHIFTS ON R0
          000014
005362 012602 SCAXIT: MOV (SP)+,R2
005364 012601          MOV (SP)+,R1

005366 000205          RTS R5
;
005370 004567 SCALO1: JSR R5,LSHIFT
          000012
005374 000772          BR SCAXIT
;
;
; RIGHT SHIFT ROUTINE

; PERFORMS (R1) RIGHT SHIFTS ON R0
; (R1) SHOULD BE GREATER THAN ZERO

005376 006200 RSHIFT: ASR R0 ;SHIFT RIGHT

005400 005301          DEC R1 ;DECREMENT COUNT
005402 001375          BNE RSHIFT ;LOOP BACK IF NOT FINISHED
005404 000205          RTS R5
;
; LEFT SHIFT ROUTINE
; PERFORMS (R1) LEFT SHIFTS ON R0

; R1 SHOULD BE GREATER THAN 0
005406 006300 LSHIFT: ASL R0 ;SHIFT LEFT
005410 102403          BVS LSHERR

```

PAGE 010

```
005412 005301      DEC R1  ;DECREMENT COUNT
005414 001374      BNE LSHIFT ;LOOP BACK IF NOT FINISHED
005416 000205      RTS R5
005420 006000 LSHERR: ROR R0      ;CRUDE ATTEMPT IN HOPE SYSTEM MAY RECOVER
005422 010446      MOV R4,-(SP)
005424 012704      MOV #LSHMES,R4
          005440
005430 0045E7*     JSR R5,MESS
          000000
```

```
005434 012604      MOV (SP)+,R4
005436 000205      RTS R5
005440 015      LSHMES: .BYTE 15,12
005441 012
005442 117          .ASCII /OV.FL:ASL/
005443 126
005444 056
005445 106
005446 114
005447 073
005450 101
005451 123
005452 114
005453 015          .BYTE 15,12,0
005454 012
005455 000
```

```
;
;
;
;
;
;NORMALIZATION ROUTINE
;CALL:
```

```
; JSR R5,NORM
;R0 CONTAINS THE FRACTION TO BE NORMALIZED
;R1 THE SCALE FACTOR OF NUMBER IN R0
;ON RETURN--R0 CONTAINS NORMALIZED FACTION
;AND R1 THE ASSOCIATED SCALE FACTOR
```

```
;
;
005455 005301 NORM: DEC R1  ;DECREMENT SCALE
005460 006300      ASL R0  ;SHIFT 0S INTO LOWER BIT
005462 001404      BEQ NFIN   ;IF RESULT IS 0,OPERATION IS COMPLETE
005464 102374      BVC NORM   ;IF SIGN DID NOT CHANGE CONTINUE
005466 006000      ROR R0      ;RESTORE SIGN
005470 005201      INC R1      ;AND SCALE
005472 000404      BR NDONE   ;NORMALIZATION COMPLETE
005474 006000 NFIN: ROR R0  ;RESTORE SIGN:000000 OR 100000
005476 006200      ASR R0      ;AND REPLICATE IT:000000 OR 140000
005500 005201      INC R1      ;INCREMENT SCALE
005502 005201      INC R1      ;INCREMENT SCALE
```

```
005504 000205 NDONE: RTS R5
;
;
;ARITHMETIC PACKAGE TEST ROUTINE
```

```

;NEEDS TTY IO PACKAGE
;ACCEPTS + INTEGER I FROM TTY AND
;PRINTS LOG10(I)*2**SCALE WHERE
;REQUIRED VALUE OF SCALE HAS TO BE SET
;IN THE APPROPRIATE VARIABLE ADDRESS

```

```

;
005506 004567 ARTEST: JSR R5,INIT
000000

```

```

005512 004567 ARBACK: JSR R5,INBUF
000000
005516 000001 .WORD 1
005520 005554 .WORD AWORD,AR1
005522 005560
005524 016700 MOV AR1,R0
000030

```

```

005530 004567 . JSR R5,SCALOG
177574
005534 010067 MOV R0,DR1
000022
005540 004567 JSR R5,ICABUF
000000
005544 000002 .WORD 2,AR1,DR1
005546 005560
005550 005562
005552 000757 BR ARBACK
005554 111 AWORD: .ASCII /I=/
005555 075
005556 000 .BYTE 0
005560 005560 .EVEN
005560 000000 AR1: .WORD 0
005562 000000 DR1: .WORD 0

```

```

;
;
005506 .END ARTEST

```

000000 ERRORS

ARBACK	005512	ARTEST	005506 G	AR1	005560
AWORD	005554	CL0	= 026501	CL1	= 110322
CL3	= 155364	CL5	= 164354	DIMES2	004747
DIV	004550 G	DIVERR	004676	DIVER2	004712
DIVLUP	004642	DIVMES	004734	DIVZER	004672
DIV1	004566	DIV2	004610	DIV3	004622
DIV4	004662	DIV5	004666	DIV6	004646
DR1	005562	FDIV	004752 G	FMULT	004776 G
GETNUM	= ***** G	ICABUF	= ***** G	INBUF	= ***** G
INIT	= ***** G	LOG	005010 G	LOGERR	005224
LOGMES	005242	LOG1	005170	LOG2	005216
LSHERR	005420	LSHIFT	005406 G	LSHMES	005440
L102	= 154570	MESS	= ***** G	MLOOP	004444
MULERR	004516	MULMES	004530	MULOUT	004472
MULT	004400 G	MULT1	004422	MULT2	004436
MULT3	004504	MULZER	004510	NDONE	005504
NFIN	005174	NOADD	004456	NORFIN	005312
NORM	005456 G	NORM0	005270	NORXIT	005326
PC	=%000007	RSHIFT	005376 G	R0	=%000000
R1	=%000001	R2	=%000002	R3	=%000003
R4	=%000004	R5	=%000005	SCALE	= ***** G
SCALOG	005330 G	SCAL01	005370	SCAXIT	005362
SP	=%000006		= 005564		

ARTEST	= ***** G	BUFEND	002520	BUFFER	002410
CRLF	002405	DELETE	002375	DIRTY	002544
DWORD	003134	EXEART	003502	EXECON	003444
EXECUT	003322	EXEDAY	003616	EXEDA1	003700
EXE IOT	003506	EXELUP	003356	EXEOUT	003432
EXERES	003476	EXETIM	003512	EXETI1	003600
EXE2	003772	EXE3	004000	EXE4	003760
EXE5	003766	EXE6	004024	EXE7	003744
GBACK	002562	GBAD	002670	GDONE1	002656
GETCHR	002124 G	GETFLG	003720	GETNUM	002554 G
GEVIL	002702	GEWAIT	002134	GLOOP1	002574
IBUFBA	003236	IBUFER	003230	ICA	003004 G
ICABUF	002716 G	ICAERR	003116	ICALUP	002734
ICA1	003062	ICA2	003046	ICA3	003030
ICEND	003076	ICLOOP	003040	INBUF	003150 G
INBU1	003164	INIT	002052 G	IOTEST	002000 G
LBIG	002370	LDEL	002404	LDONE1	002344
LDONE2	002330	LENTER	002216	LINE	002204 G
LKSERV	= ***** G	LLOOP	002232	LOK	002306
MAINBA	002004	MAINBU	002044	MAINME	002032
MDONE	002164	MESS	002150 G	MLLOOP	002152
MONDAY	= ***** G	MONDET	003722 G	PC	=%000007
PUTCHR	002170 G	RDRINT	003256	RDRLOC	003716
RDR1	003310	RUBOUT	002522	R0	=%000000
R1	=%000001	R2	=%000002	R3	=%000003
R4	=%000004	R5	=%000005	SP	=%000006
TDAY	= ***** G	THR	= ***** G	TMIN	= ***** G
TMON	= ***** G	TOOBIG	002356	TSEC	= ***** G
TWOBIG	002362	TYEAR	= ***** G	VISIB	= ***** G
WORCNT	003002		= 004034		



BALEPS	010562	BASE	010502	BGILIN	010214
BUFFIC	010470	CAID	010572	CAIK	010574
CALBAC	005642	CALC	005600	CTRFLG	010522
DIAGNO	010614	DISERR	010462	DISPLA	010332
DISPL1	010374	DISPL2	010436	DISPL3	010454
DISPL4	010354	DIV	= ***** G	DPRNT1	006376
DPRNT2	006416	DPR1	006414	DPR2	006444
DUM	010566	ET	010506	ETTBL	010270
FSTIME	010520	GETCHR	= ***** G	GETNUM	= ***** G
ICA	= ***** G	ICABUF	= ***** G	INBUF	= ***** G
INFLAG	010524	INIT	= ***** G	INMODE	010512
INPUT	007214	INTFAC	010030	INTSTY	010504
INTTBL	010260	INTTY	007230	IOD1	010126
IOD10	010200	IOD11	010236	IOD12	010246
IOD3	010072	IOD4	010064	IOD7	010134
IOD9	010170	ITCRLF	006372	ITER	005750
ITERER	006342	ITEREX	006306	ITERLU	005762
ITERMA	= 000012	ITERME	006360	ITER1	006212
ITER10	006144	ITER11	006114	ITER2	006224
ITER3	006234	ITER4	006252	ITER5	006260
ITER6	006340	ITER8	006136	ITER9	006140
ITIV	010570	ITMESS	006446	ITRANS	010554
LEPS	010624	LKSERV	006676 G	LK1	006726
LK2	006744	LK3	006762	LK4	007136
LK5	007164	LK7	007200	LK9	007116
LK9	007126	LOGNBM	010576	LSHIFT	= ***** G
LS200	010622	MAGSCA	010620	MESS	= ***** G
MEXIT	010626	MCDNR	010630	MONDAY	010546 G
MONDET	= ***** G	MULT	= ***** G	MWORD1	007706
MWORD2	007712	MWORD3	007720	MWORD4	007730
MWORD5	007734	MWORD6	007274	MWORD7	007304
MWORD8	007310	MWORD9	007320	NCOUNT	010516
NINPUT	= 000005	NOITER	010556	NPUL	010606
NTRANS	= 000002	NTRDBL	= 000004	NWORD1	007744
NWORD2	007754	NWORD3	007764	OUTATC	006532
OUTA1	006544	OUTMES	006602	OUTPUT	006504
OUTREC	006546	OUTSTR	010306	PARMES	007656
PBACK	007624	PC	=%000007	PRATE	010510
PUTCHR	= ***** G	RINTIN	010152	RSELEC	010526
RSHIFT	= ***** G	R0	=%000000	R1	=%000001
R2	=%000002	R3	=%000003	R4	=%000004
R5	=%000005	SCALE	010616 G	SCALOG	= ***** G
SP	=%000006	SPABAC	010002	SPACE	007774
SFCRLF	010024	STROBE	010300	TARSEC	010534
TCTR	010514	TDAY	010544 G	THR	010542 G
TMIN	010540 G	TMON	010550 G	TRIN	010046
TSEC	010536 G	TTYDAT	007236	TTYPAR	007330
TTYPA1	007614	TTYPA2	007650	TTYPA3	007606
TTYPA4	007442	TW	010530	TWCTR	010532
TYEAR	010552 G	V	010600	VINIT	010612
VISIB	010632 G	VISLUP	010642	VMAX	= 013560
VMIN	= 000052	VNEW	010560	VSTORE	010602
	= 010666				

## REFERENCES

1. Ingraio, H.C. and J.R. Lifnitz, "Characteristics of a Signal Data Converter for a Multi-Runway Visibility Measuring System," Report No. DOT-TSC-FAA-72-1, October 1971.
2. Ingraio, H.C. and J.R. Lifnitz, "Proposed Control Tower and Cockpit Visibility Readouts Based on an Airport-Aircraft Information Flow System," Report No. DOT-TSC-FAA-71-18, July 1971.
3. Coury, F.F., "A Practical Guide to Minicomputer Applications," IEEE Press, The Institute of Electrical and Electronics Engineers, Inc., New York NY, 1972.

## APPENDIX II

### ANALOG COMPUTER FOR CALCULATING RVR

This appendix describes the design and operation of an analog computer for calculating RVR developed by Dr. Joseph L. Horner. This work was sponsored by the Federal Aviation Agency and in compliance with the Project Plan Agreement (PPA) FAA-515.

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
1. ANALOG COMPUTER.....	II-1
1.1 Introduction.....	II-1
1.2 Analog Computation of RVR.....	II-1
1.3 Construction of Analog Computer.....	II-4
1.4 Performance.....	II-4
1.5 Conclusions.....	II-8
REFERENCES.....	II-10

## LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Page</u>
1-1. Block Diagram of the Analog RVR Computer Developed at TSC. The input, $t_b$ , Is Fed from a Transmissometer in the Field, and the <sup>b</sup> RVR Output Is Displayed on the Digital Voltmeter (DVM).....	II-3
1-2. Timing Sequence of Control Circuits on the Range Ramp Generator (A1) and Signals Given by the Solid State Switches ( $S_1, S_2$ and $S_3$ ) Shown in Figure 1-1.....	II-5
1-3. Schematic Diagram of Analog Computer Developed at TSC for Calculating RVR by Allard's Law.....	II-6
1-4. Photograph of Analog RVR Computer. Lower Card Holds the Log Amps and Multiplier, the Middle Card the Op Amps', Timing Circuits and Solid State Switches, and the Upper Card, the Op Amp and Pots for Setting the Parameters $E_t$ and $I_o$ .....	II-7
1-5. Output Error in Percent of Analog RVR Computer as Function of RVR Visibility in Feet.....	II-9



## 1. ANALOG COMPUTER

### 1.1 INTRODUCTION

Two psychophysical equations are used to compute the RVR. The first is Allard's Law:

$$E_t = \frac{I_o (t_b)^{R/b}}{R^2}, \quad (1)$$

where  $E_t$  is the illuminance threshold (a property of the eye and background lighting conditions),  $I_o$  the luminous intensity of the specific target light (the runway edgelights),  $t_b$  the atmospheric transmittance measured over a pathlength  $b$ , and  $R$  the visual range. The FAA-accepted values of  $E_t$  are 1000 mile-candles for daytime, and 2 mile-candles for nighttime.

The second is Kosmieder's law,

$$C_R = C_o (t_b)^{R/b}, \quad (2)$$

where  $C_o$  is the contrast of a target,  $C_R$  is the observed contrast, and the factor  $t_b^{R/b}$  is that described above. The limiting value of contrast threshold is taken to be 5.5% for aviation purposes.

The present FAA-computer selects the larger RVR value from equations (1) and (2) and displays the result. The computer is essentially a look-up table, where pre-computed value pairs of  $t_b$  and RVR are stored.

### 1.2 ANALOG COMPUTATION OF RVR

We begin by rewriting equation (1) in a consistent set of units:

$$E_t = \frac{I_o (t_b)^{R/b}}{(R/5280)^2}, \quad (3)$$

where  $E_t$  is in units of mile candles,  $R$  in feet, and  $I_o$  in candelas. To find the RVR, we must solve equation(3) for the transcendental variable  $R$ . Taking the common logarithm of both sides of (3) and rearranging terms;

$$\log\left(\frac{E_t}{I_o (5280)^2}\right) = \frac{R}{b} \log t_b - 2 \log R. \quad (4)$$

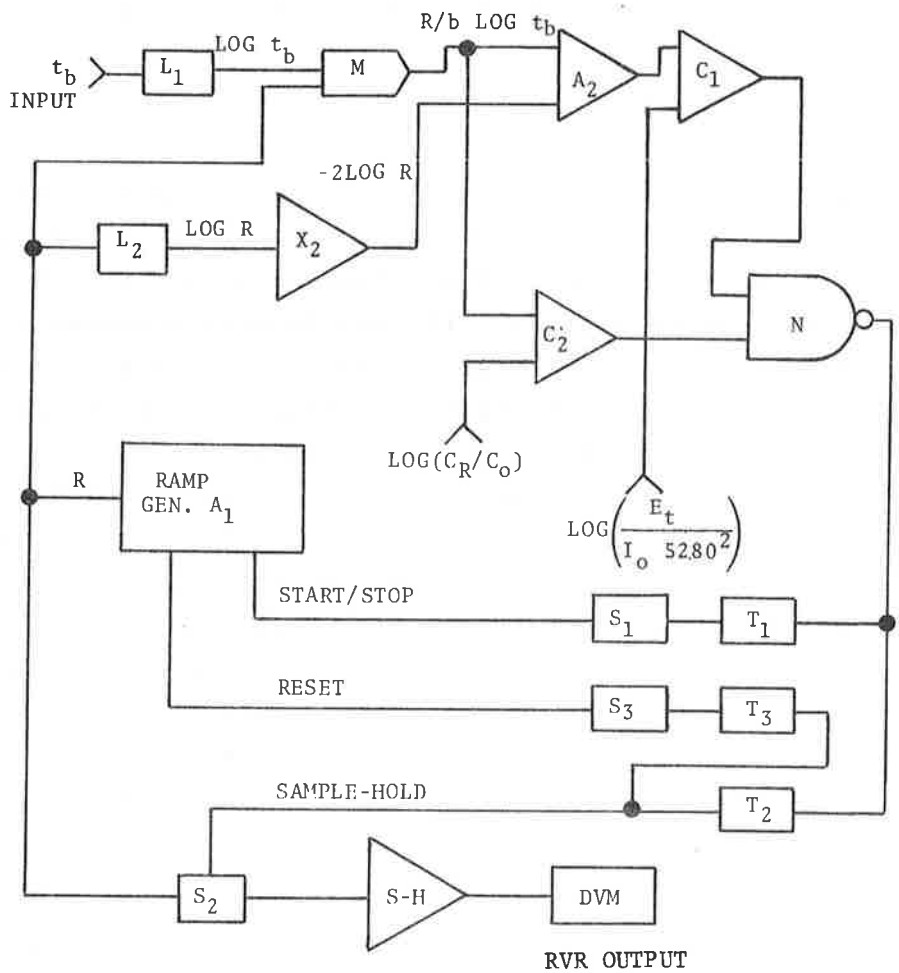
The left side contains only fixed parameters, and the right side the running variable  $R$ , and the output of the field sensor  $t_b$ , expressed as a fraction always less than 1.0. Doing the same thing for Koshmieder's Law, equation (2) with  $C_o/C_R$  set equal to 0.055 gives,

$$-1.260 = \frac{R}{b} \cdot \log t_b. \quad (5)$$

A block diagram of an analog system for solving these equations is shown in Figure 1-1. A ramp voltage is generated for  $R$  which increases linearly with time.

The two component parts of the right side of equation (4) are assembled by logarithmic amplifiers L1, L2, multiplier M, and operational amplifiers A2 and A1. Comparator C1 compares this with the left side of equation (4). This represents the Allard's Law computation. The Koshmieder computation is performed by comparator C2. The NAND circuit, N, fires after both equations have been solved, and stops the ramp generator through timer T1 and solid state switch S1, thus automatically making  $R$  correspond to the larger of the two RVR values. Timer T2, through its solid state switch S2, connects the  $R$  voltage to a sample-and-hold (S-H) circuit which in turn displays the RVR on a digital voltmeter (DVM). Timer T3 is started after the S-H circuit has acquired the new RVR value, and through S3 resets the ramp generator to a low value of range voltage. The entire





NOTE:  $T_1 = T_2 = T_3 = 555$  Timers  
 $S_1 = S_2 = S_3 =$  FLT Switches  
 $L_1 = L_2 =$  Log Amplifiers  
 $A_2 = C_2 = C_1 =$  Comparators

Figure 1-1. Block Diagram of the Analog RVR Computer Developed at TSC. The input,  $t_b$ , Is Fed From a Transmissometer in the Field, and the RVR Output Is Displayed on the Digital Voltmeter (DVM).

cycle then repeats to find a new value of R. The timing sequence is shown diagrammatically in Figure J-2, and takes about one second for the complete sequence.

### 1.3 CONSTRUCTION OF ANALOG COMPUTER

The schematic diagram is shown in Figure 1-3, and a photograph of the unit in Figure 1-4. It is built on three plug-in circuit cards modules for ease in replacement. The bottom card carries the logarithmic amplifiers and multiplier. The middle deck contains the operational amplifiers which make up the various comparators, adders, and gain amplifiers. It also contains the timing circuits and their associated solid state switches. The upper card contains the potentiometers and adders for composing the left side of equation (4). Airports use three different runway edge-light intensity settings; 10,000, 2,000, and 400 candelas. Switches on this card allow the operator to select the appropriate light setting, as well as a day or night value for  $E_t$ . In an actual installation, this switch would interface with the tower lighting controls so that the light setting would automatically be transferred to the computer. The day/night switch would also be automated by a simple photocell and amplifier looking out the tower window.

The three cards are hand wired, but with printed circuit techniques it would be entirely possible to assemble the whole analog computer on one 5" x7" card. Because we only wanted to demonstrate feasibility, the Koshmieder equation solver, C2 and N, are not included in the package. This could be included and should present no problem as it uses quite straightforward circuitry.

### 1.4 PERFORMANCE

The range voltage R is scaled to 500 feet/volt. Since the operational amplifiers saturate at 10 volts, the maximum RVR computable is 5000 feet. The 50% voltage divider between the S-H circuit and DVM make the display read RVR directly in hundreds of feet. The lower limit in the present FAA transmissometer is a RVR of 600 feet. The chief limiting factor on the lower end of the RVR scale

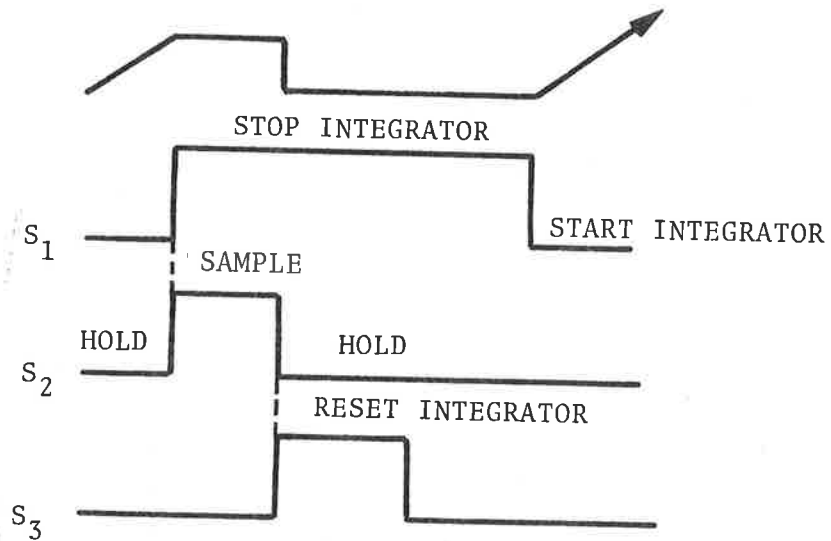


Figure 1-2. Timing Sequence of Control Circuits on the Range Ramp Generator (A1) and Signals Given by the Solid State Switches (S<sub>1</sub>, S<sub>2</sub> and S<sub>3</sub>) shown in Figure 1-1.

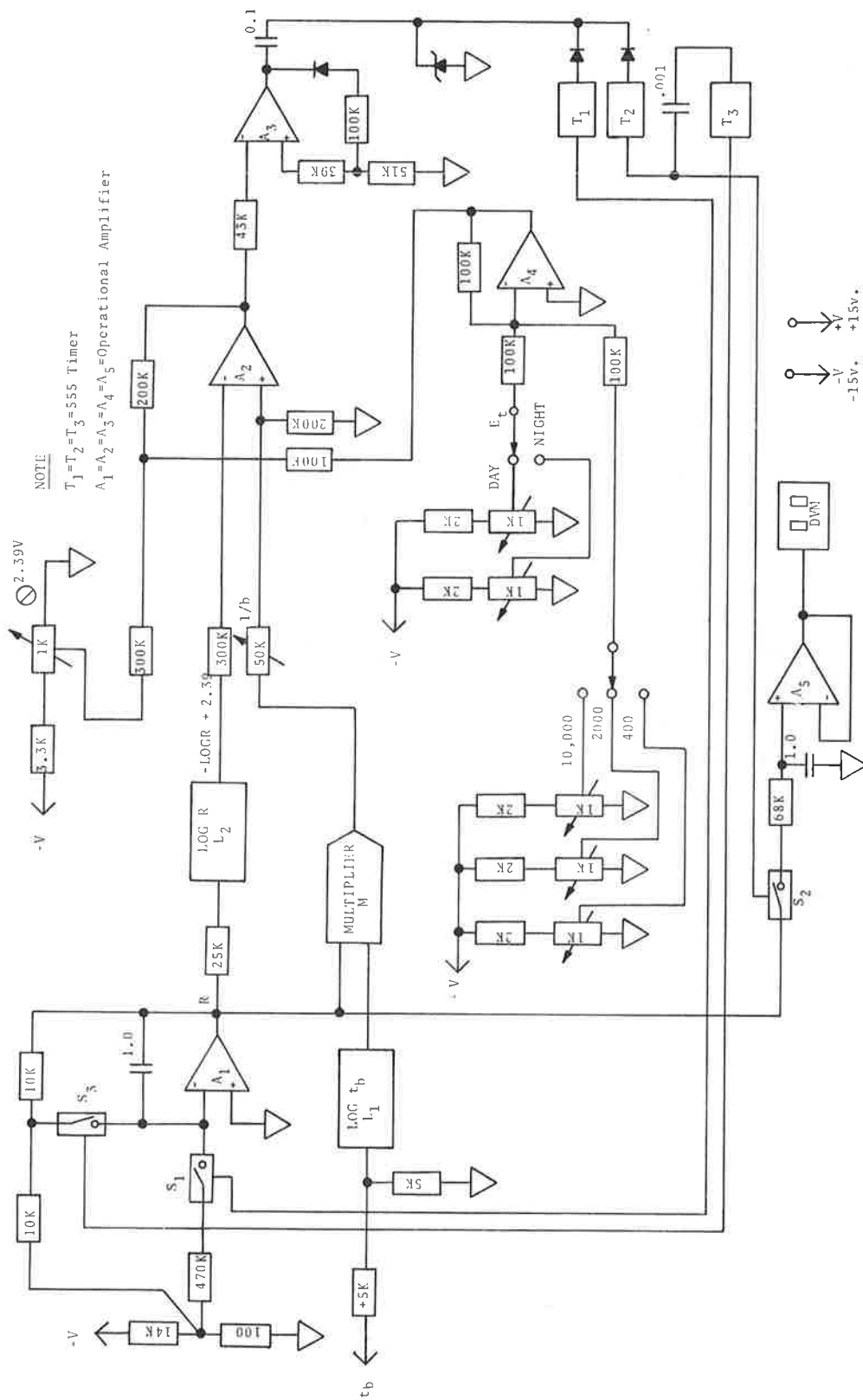


Figure 1-3. Schematic Diagram of Analog Computer Developed at TSC for Calculating RVR by Allard's Law.

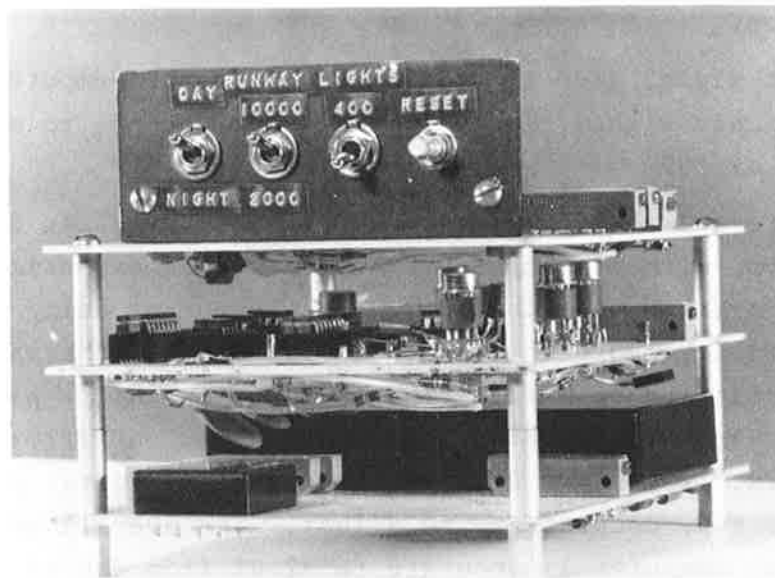


Figure 1-4. Photograph of Analog RVR Computer. Lower Card Holds the Log Amps and Multiplier, the Middle Card the Op Amps', Timing Circuits and Solid State Switches, and the Upper Card, the Op Amp and Pots for Setting the Parameters  $E_t$  and  $I_o$ .

in the analog computer is the logarithmic amplifier, L1, which computer  $\log t_b$ . This particular amplifier, according to its manufacturer, has an accuracy of 0.2% over four decades of input signal and a 1.0% accuracy over six decades. Four decades ( $t_b = 10^{-4}$ ) would correspond to a RVR of 233 feet, and six decades ( $t_b = 10^{-6}$ ) corresponds to a RVR of 167 feet (daylight conditions and runway light settings of 10,000 candelas). Therefore, it can be seen that the analog computer can easily meet a requirement of 600 foot minimum RVR computation.

The input signal from the field sensor to the computer is a 10 volt full scale signal, i.e., 10 volts corresponds to a baseline transmittance of 100%.

One can regard the visibility computation device as a six-fold curve generator, with the input variable being atmospheric transmittance over a fixed baseline ( $t_b$ ), and the output being the RVR in feet; six-fold because there are two illuminance threshold parameters in combination with three runway light intensity parameters. Figure 1-5 shows that typical analog computer output error as a function of the RVR. The exact RVR was taken from a digital computer calculation of RVR, accurate to  $\pm 0.5$  feet. The overall result is that the analog computer is accurate to  $\pm 2.0\%$  from 500 to 5000 feet (daylight conditions and 10,000 caldela runway light setting).

## 1.5 CONCLUSIONS

The device should be quite reliable, since it is all solid state with no moving parts. The temperature stability should be quite good, although this was not tested in the laboratory. The gain and reference current drift for the logarithmic amplifiers are  $\pm 0.05\%/^{\circ}\text{C}$  and  $0.1\%/^{\circ}\text{C}$  respectively.

A limitation of this analog computer in its present configuration is that it does not provide RVR values in incremental steps as required by the FAA and ICAO. The FAA transmissometer reports in increments of 200 feet between 600 and 3000 feet and increments of 500 feet between 3000 and 6000 feet.

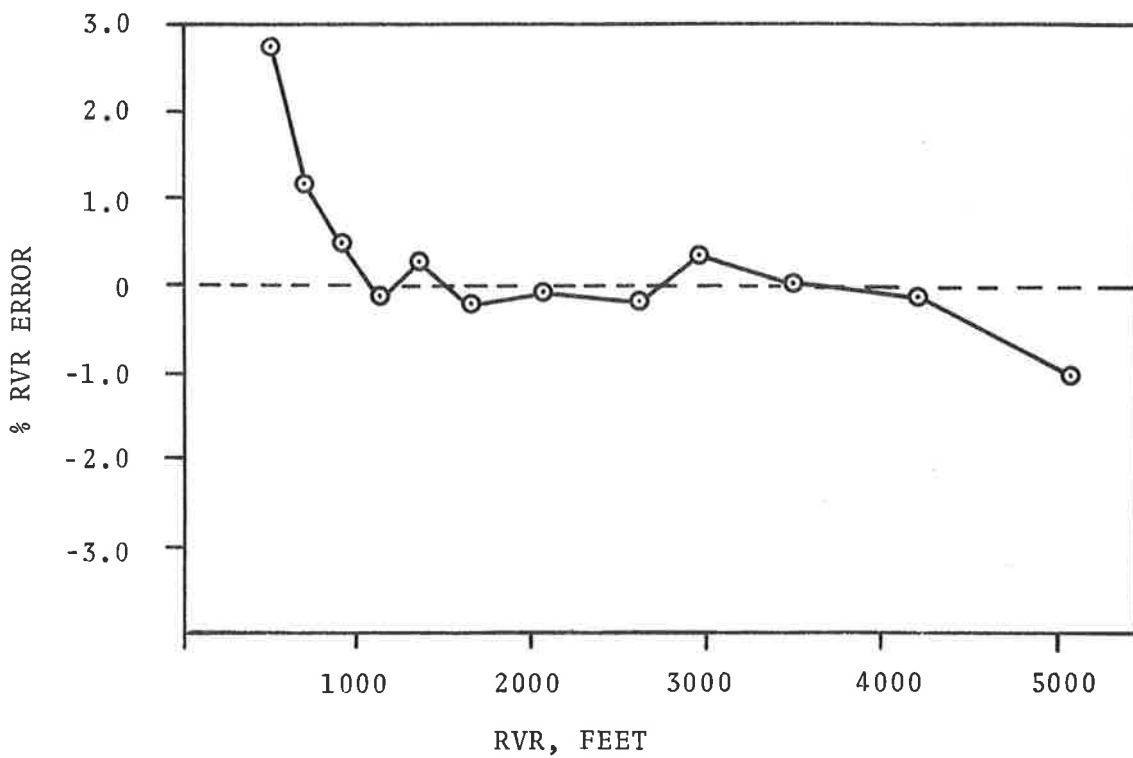


Figure 1-5. Output Error in Percent of Analog RVR Computer as Function of RVR Visibility in Feet.





1875

1876

1877

