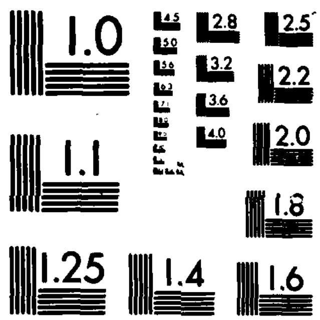
AD-A164 359 COMPUTER MODELING OF IR ENCOUNTERS AND COUNTERMEASURE EFFECTIVENESS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J W WILSON DEC 85 1/2

UNCLASSIFIED F/G 17/4 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

COMPUTER MODELING OF IR ENCOUNTERS
AND COUNTERMEASURE EFFECTIVENESS

by

Jeffery Wade Wilson

December, 1985

Thesis Advisor: Alfred W. M. Cooper

Approved for public release, distribution unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A164359	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Modeling of IR Encounters and Countermeasure Effectiveness		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1985
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) LT Jeffery Wade Wilson, USN		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93943		12. REPORT DATE December 1985
		13. NUMBER OF PAGES 101
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Atmospheric optical radiation propagation, LOWTRAN, FASCODE, DMAD, atmospheric transmittance, atmospheric radiance

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two computer modeling programs, LOWTRAN6 and DMAD have been converted for use at the Naval Postgraduate School. LOWTRAN6, used to model the propagation of optical radiation through the atmosphere at low resolution, has been modified to permit interactive use on the school's IBM 3033AP computer; two additional programs have been developed to provide interactive data acquisition and plotting.

DMAD has been implemented on a VAX-11/780 computer; it is used -

Block 20 (Cont.):

to model the effectiveness of infrared countermeasure decoys against infrared-seeking anti-ship missiles.

Work has also been conducted toward the conversion of FASCODE, an infinite resolution line-by-line transmittance/radiance code. The AFGL MAIN line parameters tape has been implemented on the IBM 3033AP.

Many computer programs presently used in the mathematical modeling of weapons and combat systems ignore some critical physical parameters. Using LOWTRAN6 data, the importance of including atmospheric parameters such as transmittance in a model such as DMAD has been demonstrated.

Approved for public release; distribution is unlimited

Computer Modeling of IR Encounters and Countermeasure Effectiveness

by

Jeffery Wade Wilson
Lieutenant, United States Navy
B. S., Oregon State University, 1978

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN PHYSICS

from the

NAVAL POSTGRADUATE SCHOOL
December 1985

Author: Jeffery Wade Wilson

Approved by: Alfred W. W. Cooper, Thesis Advisor

Eugene C. Crittenden Jr., Second Reader

Gordon E. Schacher, Chairman,
Department of Physics

John N. Dyer,
Dean of Science and Engineering

ABSTRACT

Two computer modeling programs, LOWTRAN6 and DMAD have been converted for use at the Naval Postgraduate School. LOWTRAN6, used to model the propagation of optical radiation through the atmosphere at low resolution, has been modified to permit interactive use on the school's IBM 3033AP computer; two additional programs have been developed to provide interactive data acquisition and plotting.

DMAD has been implemented on a VAX-11/780 computer; it is used to model the effectiveness of infrared countermeasure decoys against infrared-seeking anti-ship missiles.

Work has also been conducted toward the conversion of FASCODE, an infinite resolution line-by-line transmittance/radiance code. The AFGL MAIN line parameters tape has been implemented on the IBM 3033AP.

Many computer programs presently used in the mathematical modeling of weapons and combat systems ignore some critical physical parameters. Using LOWTRAN6 data, the importance of including atmospheric parameters such as transmittance in a model such as DMAD has been demonstrated.

TABLE OF CONTENTS

I. INTRODUCTION

A computer model is a mathematical representation of a physical system; such models are finding ever increasing use in the scientific and engineering communities. A realistic model allows a researcher to develop and test theories in situations where resources are scarce, the environment is hostile or the equipment involved is expensive. For example, the cost of developing a sophisticated model and using it to simulate several thousand encounters between a ship and an infrared-homing anti-ship missile (IRASM) is dwarfed by the expense of a single live shot.

Utilizing computer models, an experiment can be carefully designed to maximize the use of precious resources. Empirical data from such an experiment, when thoroughly and thoughtfully analyzed, reinforces or rejects the hypotheses upon which the model is based. This feedback system provides the mechanism by which models are improved.

The Naval Postgraduate School has an ongoing analytic and experimental program of study in the area of optical (visible and infrared) radiation propagation through the atmosphere. In support of this effort, this project was initiated to study and implement computer modeling and simulation programs and to analyze them with an eye toward possible refinement or enhancement.

Over the eighteen month lifetime of this project, the direction of research has changed many times. Initially, the goal was to implement several computer simulation codes in support of the joint USAF/USN COMPASS HAMMER project, an airborne electro-optical countermeasure system. Navy funding for the COMPASS HAMMER project was, in large part, terminated; as a result, work stopped on the computer program implementation and conversion. The COMPASS HAMMER concept is very interesting and exciting; the technology is challenging and the ramifications of such a device upon the fleet are considerable.

Attention was then directed to the implementation of the Fast Atmospheric Signature Code (FASCODE); a predictor of atmospheric effects on extremely narrow-band visible and infrared radiation. Rather than launch into the conversion of such a large and technically complex program, it was decided to convert a conceptually smaller program, LOWTRAN6, to establish a "lessons learned" base. In physical size, LOWTRAN6 and FASCODE 1C are very similar; however, the mathematics involved in LOWTRAN6 are much more manageable for a first effort.

The LOWTRAN6 model is particularly useful when a rapid, broadband, relatively low resolution calculation of atmospheric transmittance and radiance is required. An example of such a case is the infrared range equation trade-off analysis involved in the design of an infrared imager; LOWTRAN6 provides the average transmittance or integrated radiance in a given spectral band. Another excellent usage for this program would be in student laboratory or classroom work; it could be used to estimate the signal-to-noise ratio to be expected in an imaging system, given the source-detector geometry and the parameters of the operating environment. The loss in target/background contrast could also be estimated.

A product of the Air Force Geophysical Laboratory (AFGL), LOWTRAN6 is written in a Control Data Corporation (CDC) dialect of

FORTRAN IV for use on a CDC 6600 computer. The NPS version of LOWTRAN6 consists of the original code plus several machine-dependent changes to permit its use on the school's IBM 3033AP computer system.

The LOWTRAN6 conversion proceeded well, followed by the writing of an interactive input service routine (LOWINPUT) and an extensive plotting routine (LOWPLOT). With the knowledge gained in the conversion and writing of these programs, the implementation of FASCODE 1C proceeded.

Machine dependent features in FASCODE 1C have delayed its installation at NPS, although its data files were successfully read and converted for use later by student and faculty researchers.

A significant opportunity to use the knowledge gained in the study of the LOWTRAN6 algorithm arose in the Summer of 1984 with the establishment of the Naval Academic Center for Infrared Technology (NACIT) at the Naval Postgraduate School. The center was commissioned by the Naval Electronic Systems Command (NAVELEX) to install and study the Advanced Development Model (ADM) of the AN/SAR-8 Infrared Search/Track (IRST) device. LOWTRAN6 is directly applicable to the analysis and modeling of the SAR-8 as the device is a broadband infrared imager designed to operate in a marine environment. Installation of the SAR-8 is scheduled for the second quarter of FY85.

Also during this time frame, NAVELEX commissioned work involving the conversion and implementation of DMAD, a program to model the effectiveness of countermeasure decoys against infrared-homing anti-ship missiles. The number of such missiles deployed in navies around the world has rapidly increased; NAVELEX desired to have the program available at NPS for further study and analysis. The implementation and enhancement of this program continues. The DMAD model presently ignores many characteristics of the environment, the target and the missile; this report will center on one of those perceived deficiencies, the program's assumption of a nonattenuating medium for optical transmission. Other shortfalls will be treated briefly.

We begin this report with a short discussion of previous efforts conducted at the Naval Postgraduate School in the area of atmospheric optical propagation modeling. We follow this with a brief discussion of each of the three models, paying special attention to those portions of LOWTRAN6 that should be extracted and implemented in the DMAD model to correct areas of model over-simplification.

We then discuss the passive infrared range equation, examining its constituent parameters. We will devote special attention to the atmospheric transmittance term, as it is the parameter computed for us by programs such as LOWTRAN6 and FASCODE. As mentioned before, the DMAD program, in computing the probabilities of a ship-missile encounter, assumes an artificially benign propagation medium. We will explore the ramifications of such an assumption through numerical and graphical output obtained by running the NPS version of the LOWTRAN6 model (henceforth referred to as NPS LOWTRAN6).

We will close with specific recommendations for improvements in each of the three computer models discussed.

The programs written as part of this project are available upon request from the Naval Academic Center for Infrared Technology, Code 61Cr, Naval Postgraduate School, Monterey, CA, 93943. Only the shorter

EXEC2 programs are listed here; those for the remaining programs are not provided as they would add approximately five hundred pages to this report.

II. PREVIOUS WORK

The LOWTRAN low resolution atmospheric propagation prediction code has been developed and improved by AFGL since the early 1970's. In 1983, Shin [Ref. 1] translated and adapted a version of LOWTRAN 3B for use on the NPS IBM 3033AP computer; this particular version had been obtained from NWC China Lake where it had been implemented on a UNIVAC 1110. Shin's work made available an IBM-compatible low resolution atmospheric propagation model at NPS. Shin applied this model to a comparison of predicted and empirical transmittance data made at San Nicolas Island; he concluded that LOWTRAN 3B significantly overestimated optical transmittance. He obtained similar results in a comparison of LOWTRAN 3B data with that obtained during transmittance measurements over Monterey Bay.

While LOWTRAN is a suitable prediction code for broadband radiation, a high resolution, line-by-line (HITRAN) code is required for modeling and analysis of laser systems. Such a code was the AFGL computer model LASER [Ref. 2]. In 1979, Guner [Ref. 3] studied the feasibility of adapting LASER for use at NPS and concluded that the complexity of the translation made development of a new code a more desirable alternative. He wrote such a program and applied it to a Monterey Bay climatology to provide a local predictive database.

Implementation of LASER was re-initiated in 1983, as part of this project, but was abandoned when the new version of FASCODE, FASCODE 1C was received; FASCODE 1C and its database, the AFGL MAIN line parameters tape, superseded the LASER code.

DMAD is an infrared missile-ship simulation program developed at NRL by Calomiris et al. It was written to operate on a PRIME minicomputer and is highly machine-dependent in its original form. The work outlined below represents the first attempt to implement this program at the Naval Postgraduate School.

III. THE COMPUTER MODELS

A. LOWTRAN6

1. Program Description

LOWTRAN6 [Ref. 4] is a FORTRAN computer program designed to calculate atmospheric transmittance and thermal radiance for a user specified path in the spectral range 350 to 40,000 cm^{-1} (0.25 to 29 micrometers) with a 20 cm^{-1} spectral resolution. The program is essentially a computer curve-fitting program relating empirical attenuation data to a few simple meteorological input parameters. LOWTRAN6 uses a single-parameter band model for molecular aerosol absorption. The effects of continuum absorption, molecular scattering and aerosol extinction on radiation propagation are included in the model.

The program is the latest edition of a series of LOWTRAN simulation programs; it is an extension to and contains all the features of the previous version, LOWTRAN5 [Ref. 5].

2. Improvements Over LOWTRAN5

Improvements include the addition of new spherical refractive geometry subroutines, the improvement of the water vapor continuum model and the addition of a solar/lunar scattered radiance model. Further code improvements include the addition of a wind-dependent maritime aerosol model, a vertical structure aerosol model, a cirrus cloud model and a rain model.

a. Air Mass Computation (Spherical Refractive Geometry)

The LOWTRAN model treats the atmosphere as a set of spherically symmetric shells. LOWTRAN5 assumed a constant index of refraction between atmospheric layer boundaries; LOWTRAN6 uses a continuous profile for refractive index, with an exponential profile between layer boundaries.

Temperature, pressure and absorber (both gas and aerosol) densities are specified at layer boundaries. The temperature profile is assumed to be linear between boundaries while pressure and density profiles are exponential.

The variation of refractive index with altitude causes the bending of raypaths as they transit the atmosphere. Clearly, such an effect should be modeled in any program, such as DMAD, that is dependent upon raypath geometry.

b. Water Vapor Continuum

The water vapor continuum model originally developed for FASCODE has been implemented in LOWTRAN6. It is a significant improvement over LOWTRAN5, particularly in the 4.5 to 5.0 micrometer region.

c. Solar/Lunar Single Scattering Model

Previous versions of LOWTRAN treated scattering as a loss mechanism only. In reality, the scattering of solar/lunar radiation into the line-of-sight increases the path radiance. This addition to LOWTRAN6 addresses this problem.

d. Navy Maritime Aerosol Model

This model, developed by S. G. Gathman at the Naval Research Laboratory, postulates the existence of three components in the marine boundary layer aerosol population, each with a log-normal size distribution. The smallest is the background aerosol and is termed the "continental" component. Relatively independent of current wind conditions, it is sensitive to the amount of time the air mass has spent at sea. A user-supplied subjective measure characterizes the nature of the air mass (i. e. open ocean, continental or some intermediate value). The second component is termed the "stationary" component. It is dependent on wind history and represents aerosols formed by high winds and whitecap conditions. This component is characterized by the average wind speed over the previous twenty-four hours. The final component, the "fresh" component, is characterized by the current wind speed; it consists of whitecap-produced droplets.

e. Army Vertical Structure Algorithm

The Army vertical structure algorithm is a description of the vertical distribution of aerosols within the lowest two kilometers of the atmosphere. The parameters of this model are surface visibility, cloud ceiling height, thickness of cloud or fog layer, inversion or boundary layer height and the type of aerosol attenuation in effect. VSA is based primarily on data obtained during studies conducted on the Northern German plain, an area of considerable interest to the U. S. Army.

f. Cirrus Cloud Model

The addition of this model permits the calculation of extinction due to cirrus clouds. The model uses the cirrus thickness and cirrus base altitude as parameters; an option permits the use of random cirrus thicknesses.

g. Rain Model

The Marshall-Palmer raindrop size distribution and Mie scattering theory define the rain-induced extinction coefficient. Throughout the visible and IR windows, the rain extinction, for a given rain-rate, has been shown to be a function of path length only.

3. Model Parameters

Six atmospheric models are available: tropical (15° N), midlatitude summer (45° N, July), midlatitude winter (45° N, January), subarctic summer (60° N, July), subarctic winter (60° N, January) and the 1962 U. S. Standard atmosphere. If desired, a user may provide a new model atmosphere (or radiosonde data). For horizontal paths, meteorological data may substitute for the atmospheric model. The atmospheric models specify the altitude dependence of atmospheric pressure, temperature, water vapor density and ozone density.

One of three path types may be chosen for a particular simulation: horizontal (constant pressure), vertical or slant path between two altitudes, or vertical or slant path to space.

Four modes of operation are available; the program may calculate transmittance, atmospheric radiance, atmospheric radiance with added single-scattered solar/lunar radiance, and directly transmitted solar irradiance.

There are nine aerosol models in LOWTRAN6. They are: rural (vis. = 23 km), rural (vis. = 5 km), navy maritime (computes visibility), maritime (vis. = 23 km), urban (vis. = 5 km), tropospheric (vis. = 50 km), advection fog (vis. = 0.2 km), radiation fog (vis. = 0.5 km) and a user defined model (vis. = 23 km). Each of the above listed visibilities is a default value; it may be changed by the user.

Many other parameters of the environment to be modeled may be specified when the program is executed. One of the major benefits of this program is that it may be run many times, each time varying the parameters slightly, to give the user an indication of how total transmittance or radiance depends on the input variables. The cost of making these many runs is, of course, small. A sample LOWTRAN6 session appears later in this report.

4. Adapting LOWTRAN6 For Use at NPS

a. Obtaining the Program

LOWTRAN6 is available on magnetic tape or punched cards from the National Climatic Center (NCC), Federal Building, Asheville, NC 28801. For ease of shipping, handling and storage, the magnetic tape format is preferable.

b. Reading the Tape

Files are transferred from tape to the interactive operating system, VM/CMS (Virtual Machine/Conversational Monitor System) [Ref. 6, 7, 8], using the batch operating system MVS (Multiple Virtual System) [Ref. 9]. Once the LOWTRAN6 program code has been transferred to VM/CMS, the program conversion and implementation can proceed. Appendix A contains instructions necessary to effect the transfer.

c. Code Conversion and Compilation

AFGL was very careful to adhere to the FORTRAN IV standard, ANS FORTRAN X3.9-1966, where it did not impair performance in their own operating environment. Most modifications made at NPS were minor in nature and were restricted to bringing the code in line with the standard.

Bogart [Ref. 10] gives helpful guidance on the subject of code conversion between CDC and IBM FORTRAN compilers. Appendix B outlines the steps used in making the NPS version of LOWTRAN6.

The EXEC2 program used at NPS to compile LOWTRAN6 appears in Appendix C.

5. Using LOWTRAN6

The steps involved in using LOWTRAN6 are:

a. Log-on and Setup

Log onto VM/CMS and link to the Mini-Disk containing LOWTRAN6 and its support programs. Presently, LOWTRAN6 is stored on the 0617P 195 disk. The linkage is performed by typing:

CP LINK TO 0617P 195 AS 299 RR

The password is 'LOWTRAN'. Access the disk by typing:

ACCESS 299 L

This will make the disk volume, containing LOWTRAN6 and its support programs and files, the 'L' disk. Ensure that sufficient storage space is available on the 'A' disk to permit the storage of intermediate and output data files; the 'A' disk should not be filled to over 70% of capacity when using NPS LOWTRAN6.

b. Making LOWTRAN6 Input Data Files Using LOWINPUT

LOWTRAN6, as written by AFGL, runs in a batch processing environment; data is input to the program via punched cards. With the installation of the IBM 3033AP at the Naval Postgraduate School, most computing tasks are now performed interactively under VM/CMS. The interaction and usually rapid turnaround time afforded by this system are ideal for student and research laboratory use of scientific modeling programs such as LOWTRAN6.

An input service program, LOWINPUT, was written as part of this project to afford the user the capability to run the LOWTRAN6 model easily under VM/CMS. LOWINPUT prompts the user to provide LOWTRAN6 input parameters, fully exploiting the advantages of interactive computing. LOWINPUT represents a significant time savings and improvement in the accessibility of the model; student users of the LOWTRAN6 package have found LOWINPUT to be invaluable. The prompting nature of LOWINPUT not only makes LOWTRAN6 easy to use but also reduces the amount of material that must be prepared by an instructor when using LOWTRAN6 in a class homework exercise.

(1) About LOWINPUT. LOWINPUT utilizes the IBM utility package DMS/CMS (Display Management System for CMS) [Ref. 11, 12] to perform the user interface. DMS permits the definition of screen 'panels' which may be displayed in turn to provide information to, and collect data from, the operator. LOWINPUT transfers this data to a file (LOWTINP DATA) for later use as input to LOWTRAN6. Three panel

handling routines, written by programmers at the NPS Computer Center, supplement those provided by IBM; they are discussed in Appendix F.

(2) Running LOWINPUT. The file LOWINPUT EXEC (described in Appendix D) resides on the LOWTRAN6 disk; when executed, it links the user to the disk containing the support programs for the Display Management System (DMS/CMS), opens the file used to store the input data for LOWTRAN6 and runs the FORTRAN program LOWINPUT. To execute this file, type its name: 'LOWINPUT'.

(3) Interacting With LOWINPUT. LOWINPUT was designed to make the use of LOWTRAN6 as easy and trauma-free as possible. The user is prompted for each applicable data item; ample error checking is employed in most panels to catch obvious logic errors such as entering letters when numbers are expected. Several of the more catastrophic errors cannot be anticipated; the error obtained by entering an erroneous range of 13000 meters instead of the correct 12000 meters cannot be detected by the program. If LOWINPUT detects an input error it will request that the correct data be entered. If the input error is not detected, the user may continue with this run and later neglect the results computed by LOWTRAN6 with this data set or terminate this run of LOWINPUT and begin again. For this reason, first time users are encouraged to keep the number of data sets entered per LOWINPUT run few in number.

(4) Input Data Needed. The LOWTRAN6 technical report contains basic information concerning the input parameters for the program; LOWINPUT will prompt the user for the values of these parameters. Make sure every question on a panel has been answered

before hitting the ENTER key. Default values are provided for many data items; if that value is acceptable, simply skip that item without typing a new number.

c. Running LOWTRAN6

Run the program by typing 'LOWTRAN'; this causes the execution of the EXEC2 file described in Appendix E.

d. Plotting the Results of a LOWTRAN6 Run

The plotting program LOWPLOT can be run to produce a graphical representation of the LOWTRAN6 output.

LOWPLOT was written, as part of this project, to provide graphical interpretation for results calculated by LOWTRAN6; the program may be used to generate LOWTRAN6 output graphs quickly and easily. Like LOWINPUT, LOWPLOT was written to exploit the user interaction afforded by the use of the VM/CMS operating system.

LOWPLOT queries the user for each of the parameters necessary to specify a given graph; plotting is accomplished using the Display Integrated Software System and Plotting Language (DISSPLA) [Ref. 13, 14]. Further information about LOWPLOT may be found in Appendix G.

B. FASCODE

LOWTRAN was designed to provide computationally rapid results with rather low (20 cm^{-1}) spectral resolution. This level of performance continues to be adequate for broadband applications such as an IR range equation analysis for an imaging system. A quasi-monochromatic source, such as a laser, however, requires a much more intensive analysis of the physics involved in the propagation of the radiation through the atmosphere. This is because the radiation line width is small compared with the narrow molecular absorption line widths. FASCODE was developed to provide the necessary level of analysis.

FASCODE [Ref. 15, 16, 17] is a fast multilayer transmittance and radiance code. Often called an "infinite resolution code", the program calculates transmittance or radiance by performing a line-by-line convolution of all spectral lines (assuming a Voigt line shape) within sixty-four line half-widths of a given wavenumber. The accuracy afforded by the use of this model is not obtained without penalty, however. The computation involved in FASCODE is very expensive in terms of computer CPU time.

The Voigt line shape is used in FASCODE; this is a compromise between the pressure broadened profile (Lorentz) and the thermally broadened profile (Doppler). The Voigt profile is obtained by means of a weighted sum of the Doppler function and the Lorentz function. This approximate Voigt profile is convolved with empirical spectral line data. Calculations are performed out to sixty-four line half-widths. The spectral line data are obtained from the AFGL line parameters compilation which is discussed below.

Like LOWTRAN6, FASCODE approximates the terrestrial atmosphere by the use of concentric spherical shells, the number of which is dependent upon path geometry. Spectral absorptance is computed for each layer; the results are merged to yield total atmospheric absorptance.

Implementation of FASCODE has not yet been completed; work remains to be done in the area of modifying some machine dependent code within the program.

1. Obtaining and Reading the Program

The copy of FASCODE 1C used at NPS was obtained from a working copy at the Naval Environmental Prediction Research Facility (NEPRF), Monterey, CA. It was read and the contents transferred to the IBM 3033AP using procedures similar to those found in Appendix A.

2. Converting FASCODE

Converting the program syntax from CDC to IBM FORTRAN was not difficult; it proceeded in much the same way as the conversion of LOWTRAN. The implementation of FASCODE 1C on the IBM 3033AP ran into trouble because of array bound errors. In FORTRAN IV, array indices are integers; in some portions of the code, FASCODE 1C was using the integer portions of real numbers as array indices. The problem is that the integer portions of these numbers were not the same as they would have been if the program were running on a CDC machine. This is due to word length differences between the two computers.

3. The MAIN and TRACE Line Parameters Tapes

The AFGL atmospheric absorption line parameters compilation [Ref. 18] provides basic absorption parameters for the seven most IR-active molecular species present in the terrestrial atmosphere. This compilation is commonly referred to as the "MAIN" tape. The molecular species included are: H_2O, CO_2, O_3, N_2O, CO, CH_4 and O_2. The tape covers the spectral range 0 to 17900 cm^{-1}.

The AFGL trace gas compilation [Ref. 19] contains absorption parameters for twenty-one trace gases. The TRACE tape is particularly useful for analysis of harsh or extraterrestrial environments where these gases may be present in significant amounts.

For both the main and trace tapes, the parameters catalogued are: resonant frequency (vacuum cm^{-1}), line intensity (cm^{-1}/molecule cm^{-2} at 296K), air broadened halfwidth (HWHM) (cm^{-1}/atm), lower state energy (cm^{-1}), quantum identification, entry date, isotope and molecule code.

The tapes may be read and the data transferred to VM/CMS for use as input data for a program running interactively.

C. DMAD

1. Program Description

DMAD is a SECRET program designed to calculate the probabilities associated with the distraction of an infrared-homing anti-ship missile (IRASM) by a pattern of IR decoys launched from a targeted ship. The sections of the program dealing with target visibility are unclassified; they are the only portions of the probability algorithm discussed in this report.

The visibility of a target at sea is directly dependent on the physics of optical radiation propagation in the marine environment. The marine boundary layer, in which both missile and target reside, is very dynamic and often hostile to radiation transmission. We shall see, in the next chapter, some of the effects the environment has on the visibility of a target.

2. Program Implementation

DMAD was converted for operation on the NPS Wargaming, Analysis and Research (WARLAB) VAX-11/780 minicomputer. Notes pertinent to that process appear in Appendix H.

3. DMAD Operating Instructions

The NPS implementation of DMAD is very easy to use. Like the NPS LOWTRAN6 package, user interaction has been stressed to make the program readily accessible to users who do not have a significant computer background. DMAD, when further modified as outlined in the recommendations below, will make an excellent teaching tool in electronic warfare courses.

Complete instructions necessary for operating DMAD may be found in Appendix I. DMAD is command-driven; Appendix J contains a brief description of the commands available.

4. Current Status

As described here, DMAD runs on the NPS WARLAB VAX-11/780 and produces numerical and graphical output identical to that obtained on the PRIME minicomputer at NRL.

As presently configured, DMAD ignores target radiant intensity, sources of background radiation, atmospheric transmittance, detector sensitivity, signal processor characteristics and optical system parameters. The program assumes a clear, nonattenuating atmosphere and calculates its probabilities for the encounter based upon that assumption. We shall see, in the next chapter, the potential effects of such an assumption.

DMAD computes atmospheric refraction based upon the 4/3 earth radius model. This model, commonly used in microwave refraction, is generally considered inappropriate for the infrared spectrum.

IV. ANALYSIS

The optical geometry of an infrared-homing anti-ship missile forms a "footprint" on the sea surface; as the missile proceeds along its flight path, targets move into and out of the detector's field-of-view (FOV). In this project, we are primarily concerned with the visibility of such targets. The presence of an object in the FOV does not guarantee detection, for it may be obscured by its environment. For the period of time an object is geometrically visible we should like to know if the detector receives a spectral radiant flux (watts/micrometer) adequate to provide the necessary signal-to-noise ratio (SNR).

The problem of target visibility is particularly important in programs such as DMAD. Recall that the program assumes a nonattenuating propagation medium. Since the probability of detection is presently a function of encounter geometry alone, we are systematically over-estimating this probability. On the surface, this may seem to be a relatively harmless part of a worst-case scenario; in reality, however, because of the way in which the encounter is modeled, the visibility of the target has a major impact on the results of a model run.

We now examine the passive infrared range equation to discuss the functional relationships between its parameters. Following this we will treat, briefly, the subject of atmospheric transmittance. We conclude with a series of sample LOWTRAN6 runs; these will make evident the need for incorporating atmospheric parameters in the DMAD simulation.

A. THE PASSIVE INFRARED RANGE EQUATION

Hudson [Ref. 20] contains an excellent account of the passive IR range equation; the following is an amplification and discussion of material found there.

We begin by asserting that the detector is the limiting source of noise in the system, and that the target does not fill the instantaneous field-of-view. We may write the target's spectral radiant exitance ($W\ m^{-2}\ \mu m^{-1}$) as:

$$E_{\lambda} = \frac{I_{\lambda} \tau_a(\lambda, R)}{R^2}$$

where I_{λ} is the target's spectral radiant intensity ($W\ sr^{-1}\ \mu m^{-1}$), R is the detector-target range and $\tau_a(\lambda,R)$ is the spectral transmittance over that path. For horizontal, constant-pressure paths, we may write

$$\tau_a(\lambda, R) = e^{-\sigma R}$$

where σ is the "extinction coefficient" for the path.

In the LOWTRAN6 model, as we shall see next, the total atmospheric transmittance at a given wavelength (average over a 20 cm^{-1} interval) is the product of the average transmittances due to molecular band absorption, molecular scattering, aerosol extinction and molecular continuum absorption.

The spectral radiant power ($W\ \mu m^{-1}$) to the detector is given by

$$P_{\lambda} = \frac{I_{\lambda}\ \tau_a(\lambda, R)}{R^2}\ A_o \tau_o(\lambda)$$

A_o is the entrance aperture area of the optical system; $\tau_o(\lambda)$ is the spectral transmittance of the sensor. The spectral responsivity (V W^{-1}) of the detector is

$$R'(\lambda) = \frac{V_s}{P_\lambda}$$

where V_s is a (spectral) signal voltage. Solving for V_s (V μm^{-1}),

$$V_s(\lambda) = \frac{I_\lambda \tau_a(\lambda, R)}{R^2} A_o \tau_o(\lambda) \; R'(\lambda)$$

The total signal voltage over a spectral band of interest is given as

$$V_s = \frac{A_o}{R^2} \int_{\lambda_1}^{\lambda_2} I_\lambda \tau_a(\lambda, R) \tau_o(\lambda) \; R'(\lambda) \; d\lambda$$

From this, we obtain a signal-to-noise ratio by dividing both sides by the RMS value of detector noise.

$$\frac{V_s}{V_n} = \frac{A_o}{V_n \; R^2} \int_{\lambda_1}^{\lambda_2} I_\lambda \tau_a(\lambda, R) \tau_o(\lambda) \; R'(\lambda) \; d\lambda$$

We see clearly that, since spectral transmittance is a function of range, we cannot solve this integral in closed form. We approximate the solution of this integral by replacing the spectral quantities with their averages over the spectral bandpass of the sensor.

$$\frac{V_s}{V_n} = \frac{A_o \; \tau_a(R) \; \tau_o \; R' \; I}{V_n \; R^2}$$

We define, here, a detection threshold

$$DT = \left(\frac{V_s}{V_n}\right)^2$$

The maximum range (R_o) will be that range for which the detection threshold is achieved.

$$\frac{V_s}{V_n} = \left[DT\right]^{0.5} = \left[\frac{A_o \ \tau_a(R) \ \tau_o \ R' \ I}{V_n \ R_o^2}\right]$$

Rearranging,

$$Ro = \left[\frac{A_o \ \tau_a(R) \ \tau_o \ R' \ I}{V_n \ (DT)^{0.5}}\right]^{0.5}$$

We now substitute

$$R' = \frac{V_n \ D^*}{(A_d \ \Delta f)^{0.5}} \qquad Ao = \frac{\pi(D_o)^2}{4} \qquad A_d = \omega f^2 \qquad NA = \frac{D_o}{2f}$$

Where D^* is the detectivity, A_d is the detector area, Δf is the noise-equivalent bandwidth, D_o is the entrance aperture diameter, ω is the instantaneous field-of-view and f is the equivalent focal length of the optical system. Again rearranging and separating terms, we obtain a form of the range equation appropriate for trade-off analysis:

$$R = \left[I \ \tau_a(R)\right]^{.5} \left[\frac{\pi}{2} \ D_o \ (NA) \ \tau_o\right]^{.5} \left[D^*\right]^{.5} \left[\frac{1}{(\omega \Delta f)^{.5}(Vs/Vn)}\right]^{.5}$$

where we use the voltage signal-to-noise ratio in place of the detection threshold. The first term is concerned with target radiant intensity and atmospheric transmittance, the second contains optical system parameters, the third describes the detector and the fourth gives system characteristics.

Because of the modeling work previously done on this project (LOWTRAN6 and FASCODE 1C), the atmospheric transmittance term will be the focus of our present discussion. Obviously, however, the other terms are equally important and deserve inclusion in updated versions of the DMAD model. Representative values for each parameter may be estimated for any given modeled threat.

B. ATMOSPHERIC TRANSMITTANCE (LOWTRAN MODEL)

The reader is directed to the LOWTRAN5 technical manual for a concise, definitive discussion of the model's treatment of atmospheric transmittance.

Total atmospheric transmittance, at a given wavelength (averaged over a 20 cm^{-1} interval), is the product of the transmittances due to molecular band absorption, molecular scattering, aerosol extinction and molecular continuum absorption. The molecular band absorption is, in turn, composed of four terms, one each for the transmittances due to water vapor, ozone, nitric acid and the uniformly mixed gases (CO_2, N_2O, CH_4, CO, O_2 and N_2).

Table 1 lists the comparative importance of these transmittance constituents for four atmospheric windows. From Table 1 we see that the primary source of atmospheric extinction in the infrared region is molecular absorption.

We have seen the importance of atmospheric transmittance in the range equation and have examined the physical processes involved in extinction. We now examine a very simple scenario, utilizing NPS LOWTRAN6, to emphasize further the importance of considering this parameter in computer models such as DMAD.

Wavelength (micrometers)	Attenuation Constituents in Order of Importance
0.4 - 0.7	Aerosol Scattering Molecular Scattering Aerosol Absorption
0.7 - 1.2	Aerosol Scattering Aerosol Absorption Molecular Scattering Molecular Absorption
3.0 - 5.0	Molecular Absorption Aerosol Scattering Aerosol Absorption Molecular Scattering
8.0 - 12.0	Molecular Absorption Aerosol Absorption Aerosol Scattering

Table 1. Atmospheric Transmittance Components

C. A SIMPLE SCENARIO

Appendix K outlines the steps, to be taken by a user, to reproduce one of the sample runs discussed here. We will now make several simplifying assumptions concerning the target. The numbers chosen here have no particular significance other than being convenient to use for this example. First, we assume the ship's hot spot is a gray-body emitter at a temperature of approximately 500K and an altitude of eleven meters. From Wien's displacement law we know the peak of the emission curve will occur at approximately six micrometers. A large portion of the target spectral radiant intensity will, therefore, be located in a spectral region between eight and fifteen micrometers.

For these sample runs, we will assume the seeker contains a $Hg_{.8}Cd_{.2}Te$ detector operating at 77K. The spectral response (D^*) curve [Ref. 21] for this detector shows that sensitivity is greatest within the spectral band between approximately eight and twelve micrometers. This sensitivity range fits nicely, as we shall see, into the atmospheric transmittance "window" present between about eight and fourteen micrometers. This window is commonly called the "eight-to-fourteen band". The other transmittance window lies between approximately three and five micrometers (the "three-to-five band").

For our first example, we look at the transmittance for a sample path in the absence of aerosol attenuation. Figure 1 contains the important LOWTRAN6 parameters for this run.

Atmospheric model: Midlatitude Winter
Ray path geometry: Slant path between two altitudes
Program operation mode: Transmittance
Temperature, pressure, H_2O vapor and O_3 profiles: Normal operation
Extinction type: None
No cirrus cloud attenuation
Vertical Structure Algorithm: Not used
Initial altitude: .011 km (altitude of ship's hot spot)
Final altitude: .040 km (missile flight altitude)
Path length: 10.9 km (ship-to-missile distance)
Initial frequency: 666.667 cm^{-1} (15 micrometers)
Final frequency: 5000.000 cm^{-1} (2 micrometers)
Frequency increment: 5 cm^{-1}

Figure 1. Sample LOWTRAN6 Input Data

The results of the calculation are presented graphically in Figure 2; the data was plotted by the NPS LOWPLOT program. Notice that the 3-5 and 8-14 micrometer atmospheric windows are clearly visible. The average transmittance in the 8-12 band was determined, by the NPS LOWTRAN6 program, to be 0.5734.

The second run includes the Navy maritime aerosol model. The air mass character used was 1 (open ocean). To explore the importance of wind speed and wind history on the Navy maritime model we first obtained results with both the current wind speed and the 24-hour wind speed average set to zero. The plot of these results appears as Figure 3. Notice that there is essentially no difference between the plots in Figures 2 and 3. The 8-12 band average transmittance was 0.5724.

For the third, fourth and fifth runs, we used the values 5.0, 7.5 and 10.29 m/s respectively for both the current wind speed and the 24-hour average wind speed. The plots appear in Figures 4, 5 and 6. The average transmittances in the 8-12 band were 0.3291, 0.2006 and 0.1150, respectively. Note the ever-increasing effect wind has on the average transmittances. The wind speeds used are quite typical of surface wind conditions at sea.

For the final run, the wind speed and 24-hour average wind speed were set to zero; the rain rate was set at 1 mm/hr. Note, in Figure 7, the dramatic drop in transmittance. The average transmittance, in the 8-12 band, was 0.0086. A rain as light as 1 mm/hr caused essentially complete extinction over the selected path.

We have seen how environmental effects dramatically alter the calculated atmospheric transmittance. The DMAD model assumes unity transmittance; this assumption is clearly not valid in even the most benign of real maritime environments.

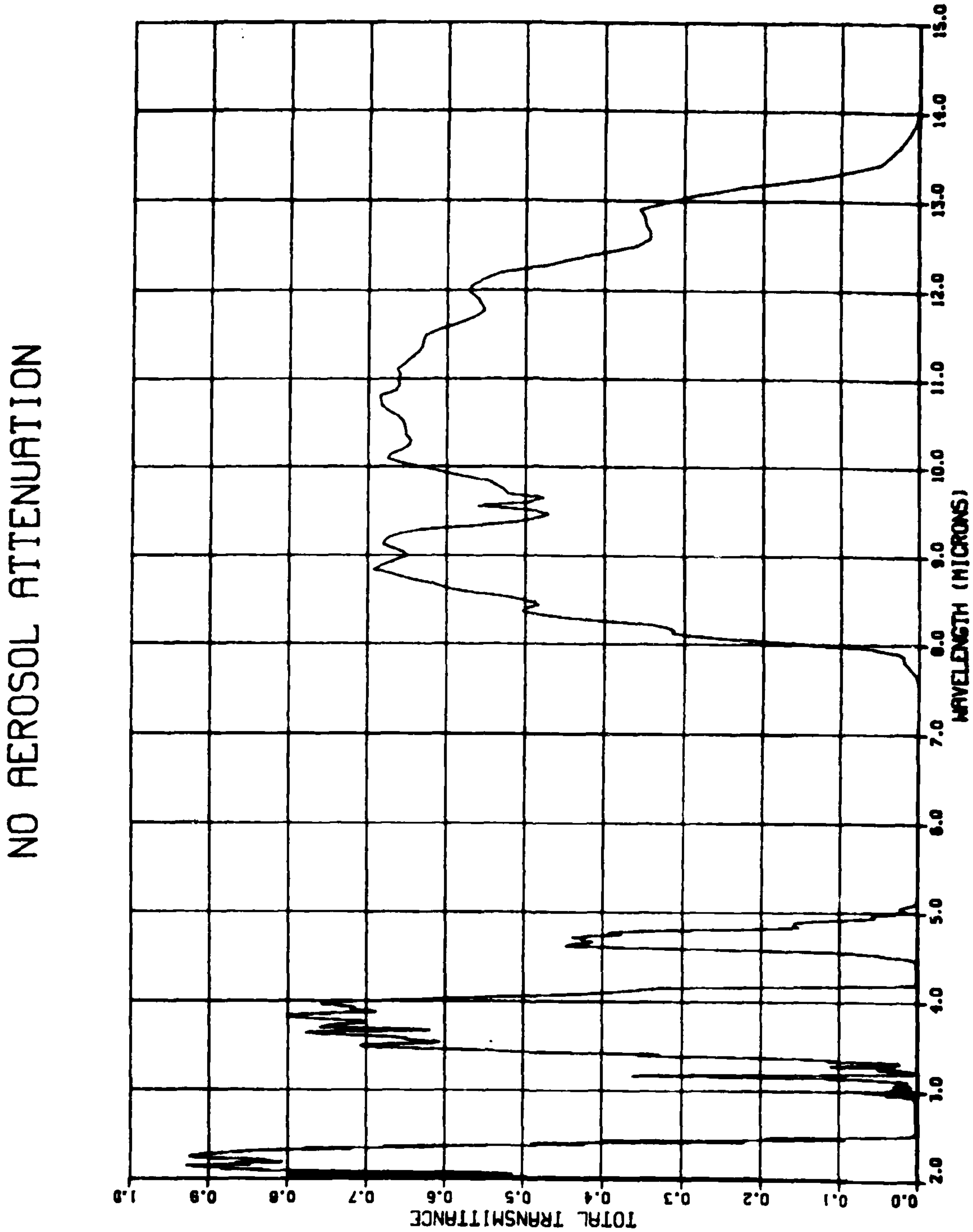


Figure 2. 10.9 km Path With No Aerosol Attenuation

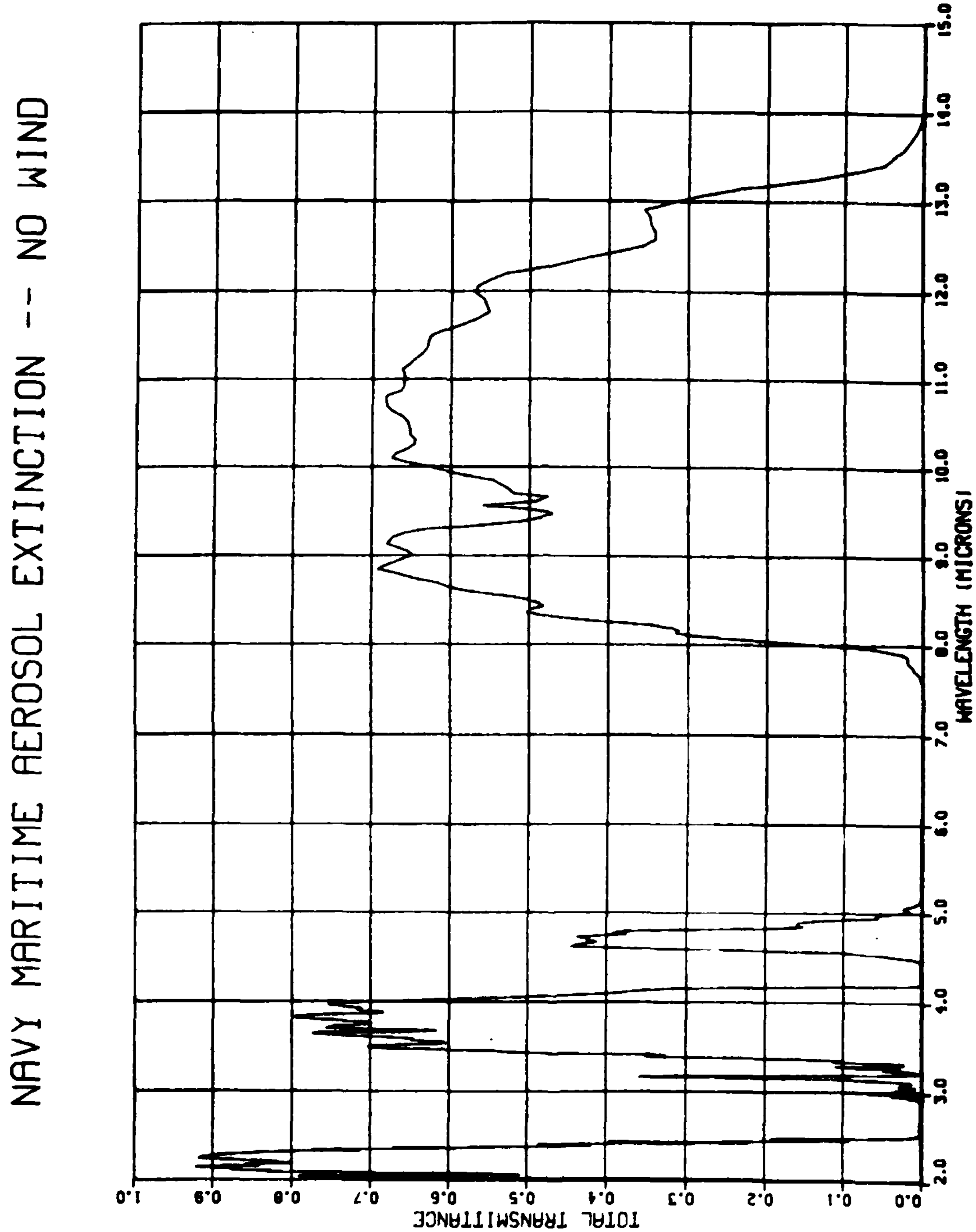


Figure 3. 10.9 km Path With Navy Maritime Aerosol Extinction (No Wind)

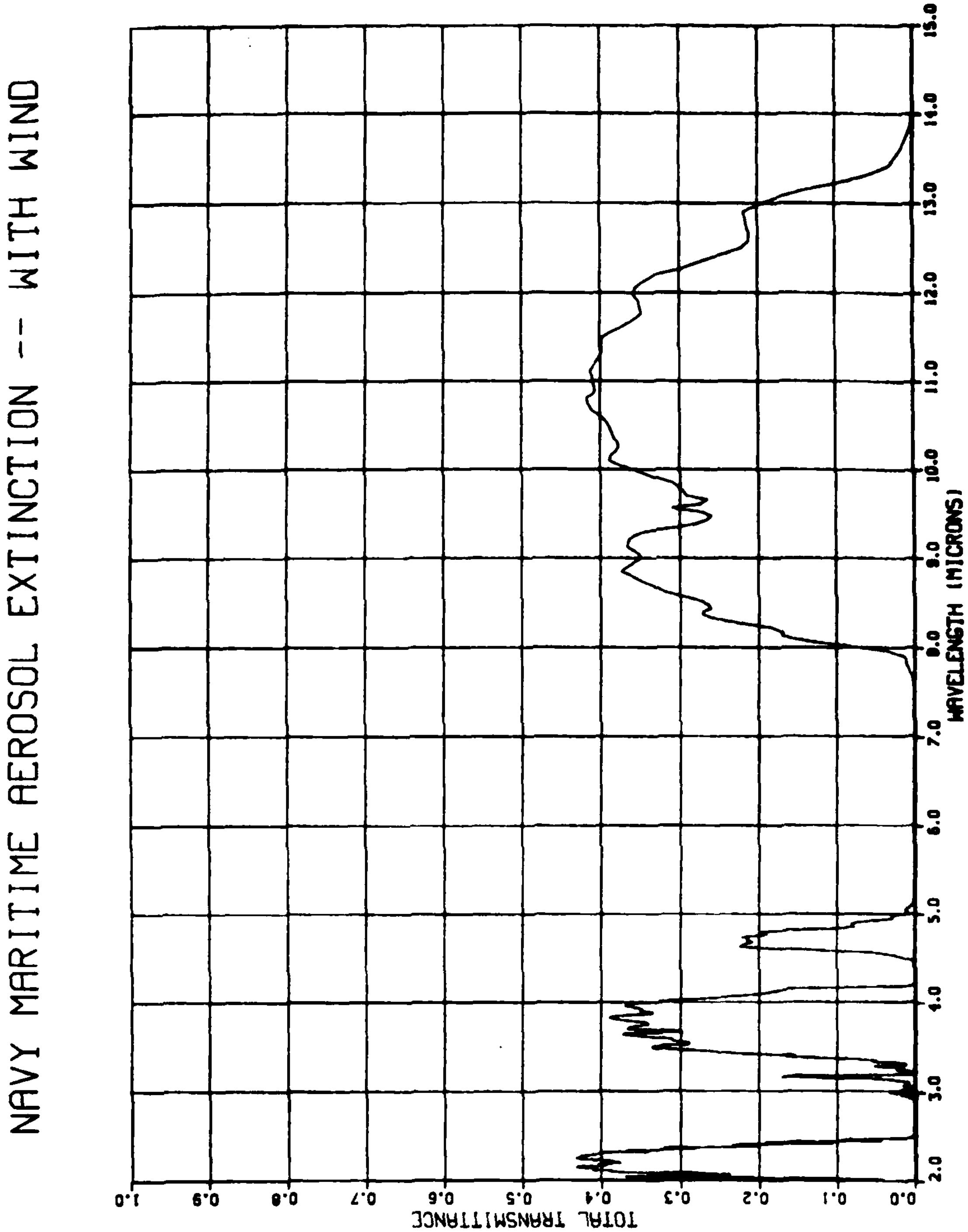


Figure 4. 10.9 km Path With
Navy Maritime Aerosol Extinction (5.0 m/s Wind

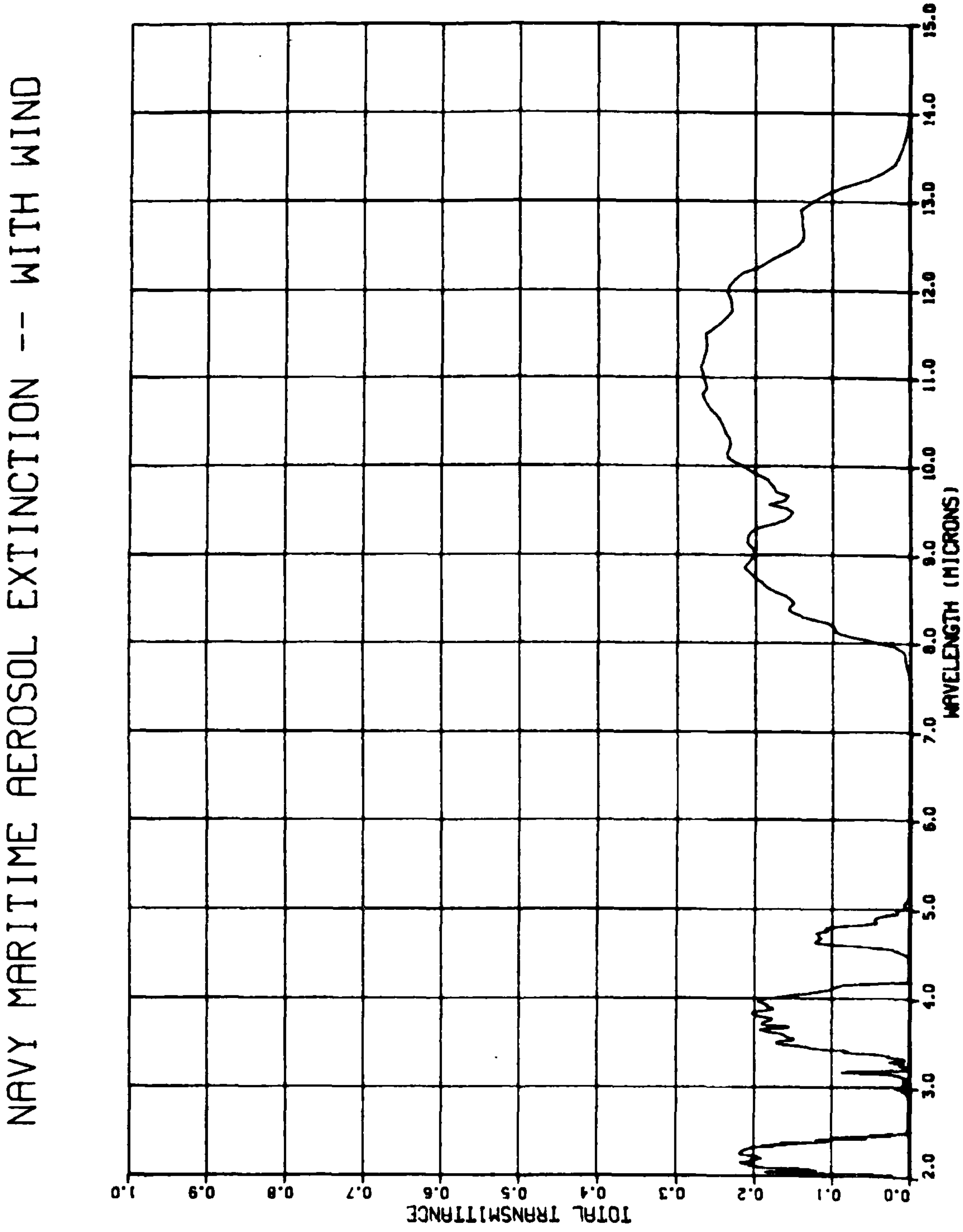


Figure 5. 10.9 km Path With
Navy Maritime Aerosol Extinction (7.5 m/s Wind)

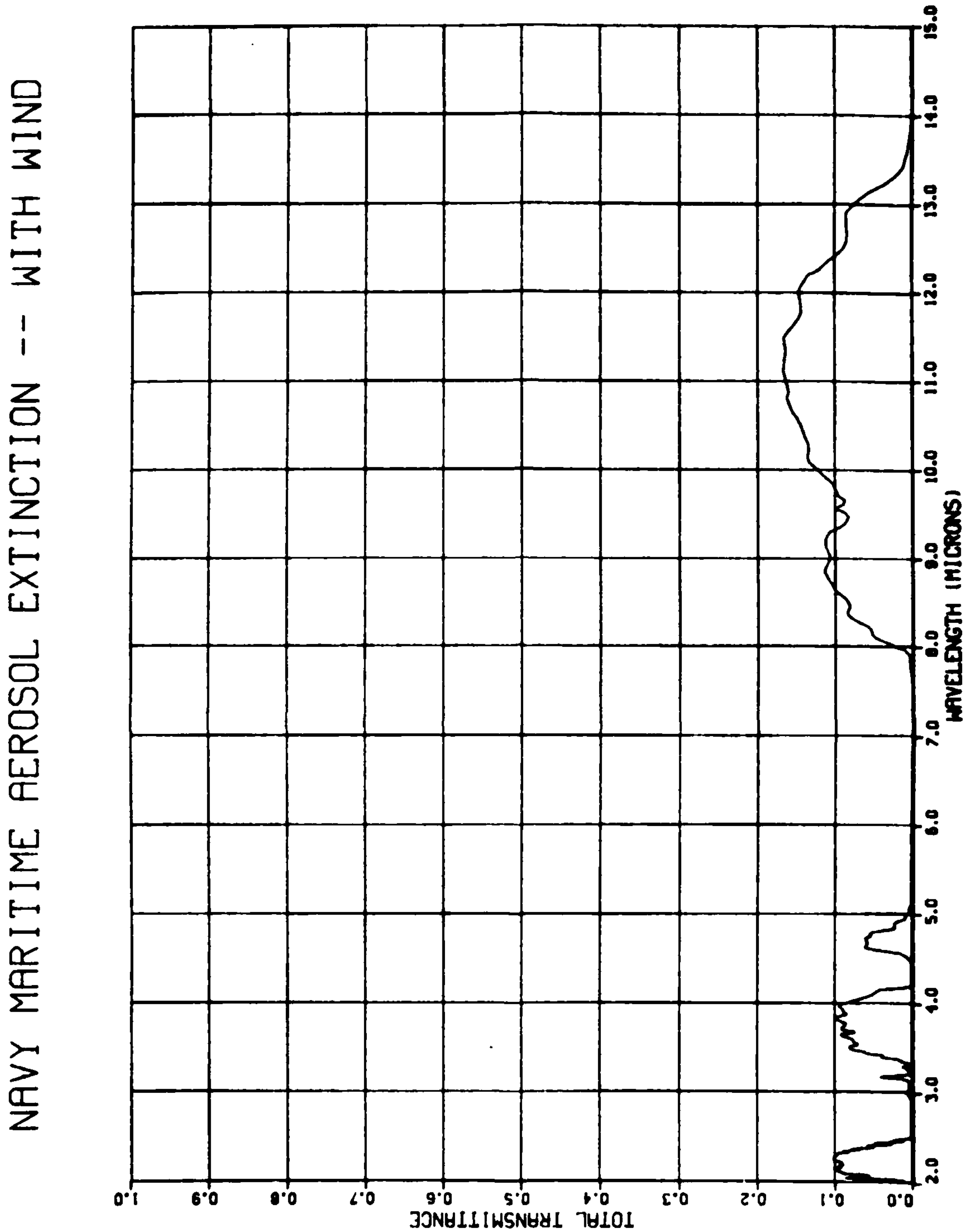


Figure 6. 10.9 km Path With
Navy Maritime Aerosol Extinction (10.29 m/s Wind)

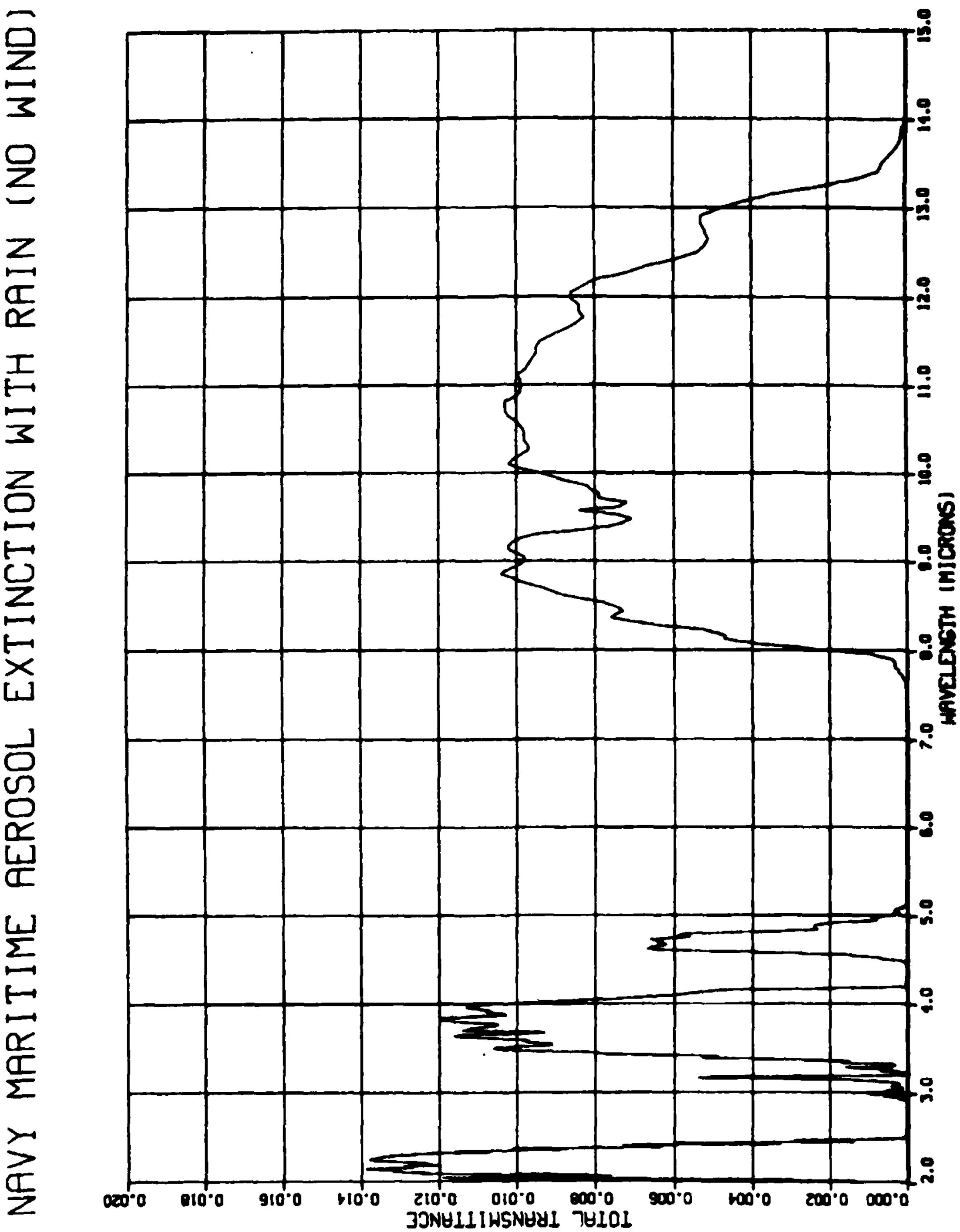


Figure 7. 10.9 km Path With
Navy Maritime Aerosol Extinction (1 mm/hr Rain, No Wind)

V. SUMMARY/CONCLUSIONS

The work undertaken on this project was interesting and generally successful. There is an enormous amount of work remaining within the scope of this project; some theoretical work, some experimental work and much computer modeling.

It is anticipated that the number and variety of computer models used at the Naval Postgraduate School will continue to increase, giving physics students an increasing opportunity to become familiar with the use of computers. Computer integration into classwork, as well as thesis research, will enhance the education of those students involved.

Three computer models were studied in this project; LOWTRAN6, FASCODE 1C and DMAD. LOWTRAN6 calculates atmospheric transmittance and thermal radiance for a user specified path. The program's effectiveness and applicability have been demonstrated through comparison of model predictions and field experimental results. LOWTRAN6 is a low resolution model (20 cm^{-1}); FASCODE 1C is an "infinite" resolution code. The validity of FASCODE 1C has also been verified by comparison of its predictions with field data. The third program, DMAD, is under development. Consequently, its effectiveness has not been tested.

A. NPS LOWTRAN6

The NPS LOWTRAN6 package (LOWINPUT, LOWTRAN6 and LOWPLOT) is a simulation tool that has wide applicability within the physics community. NPS LOWTRAN6 is easy to use and easy to maintain; these programs have been used in classwork (in the electro-optics course sequence) and in student thesis research. Most recently, the NPS LOWTRAN6 package was an integral part of the research conducted by Egolfopoulos [Ref. 22] in his analysis of ship signatures. LOWTRAN6 could also be used in system design and trade-off analysis; we have seen that transmittance plays an important role in the IR range equation.

The current plan for a follow-on to this work involves the extraction of the atmospheric refraction and transmittance subroutines for inclusion in the DMAD program. This addition will greatly enhance the realism and applicability of the DMAD model.

Another planned course of action for the near future is to convert LOWTRAN6 for operation on an IBM PC-AT microcomputer using the IBM Professional FORTRAN compiler. This move will give independence from any problems that may be generated by support program version changes and license terminations on the IBM 3033AP. The user would have more or less complete control of the hardware and software operating environment; this is important because when operating systems, languages and support packages change, significant alterations to programming packages are frequently required. An added benefit of moving to a stand-alone microcomputer system is that the computing load would be reduced on the NPS IBM 3033AP, a machine already overloaded during peak (normal working) hours.

The LOWTRAN6 program itself should not require much additional modification. LOWINPUT and LOWPLOT, however, would require extensive rewriting. Of the two, the more difficult would be LOWINPUT; efficient

FORTRAN input management routines (to replace DMS/CMS) do not presently exist for the IBM PC family. The programming involved in writing such a package is time consuming but not particularly difficult.

Converting LOWPLOT to a PC-family computer would require a great deal of effort; calls to DISSPLA subroutines would be replaced by calls to the IBM PC Plotting System, a library of FORTRAN callable routines. The effort expended in the rewriting of LOWPLOT would be more than rewarded in the greatly enhanced graphics available on the PC-AT (as compared to the IBM 3033AP installation at NPS). High resolution hardcopy plots would be drawn by a plotter such as the Hewlett-Packard 7475A instead of using an electrostatic plotter such as is used in conjunction with the TEK 618 display units.

LOWTRAN6 will not execute as rapidly on the PC-AT as it will on the 3033AP; just how much slower execution will be cannot be determined without actual experimentation. The Intel 80286/80287 microprocessor/co-processor combination present in the PC-AT make the AT one of the fastest production microcomputers today; the execution time penalty will likely not be great.

B. FASCODE

FASCODE 1C is a very important and valuable model and should be a cornerstone of IR propagation research at the Naval Postgraduate School. The implementation of the program is temporarily stalled due to errors caused by the difference between a CDC 6600 computer and the NPS IBM 3033AP machine. With effort, it is possible that FASCODE 1C can be made to run on an IBM PC-AT (with installed 80287 Numeric Data

Processor [NDP]). The IBM Professional FORTRAN compiler, available for the PC-AT, has an integral symbolic debugging facility which would make the search for array bound errors much easier.

The most severe possible drawback to the conversion of FASCODE 1C on a smaller machine is, of course, that the program may not run on a computer limited to 640K of random access memory (RAM). It will be impossible to tell if such a transfer is possible until a significant amount of effort has been expended.

Storing the MAIN and TRACE parameters on disk poses a somewhat similar problem. The MAIN tape contains approximately 181000 lines; if stored in an ASCII format, this file would consume approximately 15 megabytes (MB) of hard disk storage (the PC-AT has a 20MB hard disk). Elimination of all but essential data for each line would reduce the storage requirement to approximately 3MB, a much more manageable file size.

The AFGL MAIN line parameters tape has been implemented at NPS; if FASCODE is converted for microcomputer use, the MAIN and TRACE line parameters tapes will require further conversion and implementation.

C. DMAD

The program runs and produces output data and graphical results identical to those obtained at NRL. Several improvements to the code should be made immediately to make the program more flexible and to increase the performance and accuracy of the model. While time-consuming, none of the proposed additions/modifications to the software are particularly difficult. The proposed changes are listed below.

1. Interactive Variable Modification Facility

As the program is currently configured, a user cannot alter the values contained in program variables. At NRL, an interactive debugging facility is used to change the values of program variables while the program is running (i. e. at "run-time"). In the present NPS version of DMAD, variable changes require editing and recompilation of the code. The solution to this problem is to write a small command-driven data entry routine that would permit a user to identify the variable of interest (by typing in its name), change its value and then, if desired, re-execute the probability model to compute results based upon the new parameters.

2. Introducing Target Attributes and Atmospheric Physics

DMAD is, in many ways, a worst case scenario. By design, it systematically favors the missile in calculating the probabilities of the encounter. Nowhere is this more evident than in the areas of target description and estimation of atmospheric radiation propagation effects.

The infrared range equation, the optical counterpart to the ubiquitous radar range equation, should be considered in the complete treatment of an IRASM-ship encounter. Among the important parameters in this equation are those that describe the target. Equally important is the physics involved in the propagation of IR radiation through the atmosphere. The radiant intensity (in-band) of the source and the atmospheric transmittance (which may be calculated by LOWTRAN6) appear in the IR range equation. Currently, DMAD ignores target radiance and assumes a perfectly clear atmosphere (no attenuation). We saw, in Figures 2 through 6, examples of the dangers involved in assuming perfect transmittance.

On the surface it would seem that using an artificially high visibility would produce a strong worst-case scenario. What has not been explored yet, however, is the impact on the model's results of delaying target acquisition by realistic visibilities. Presently, DMAD calculates the encounter probabilities assuming this worst-case target acquisition range (clear air). This range, coupled with decoy deployment data, specifies the physical geometry of the encounter. Clearly, if the range is artificially high, we are using the wrong missile-decoy-ship geometry in the calculations. This may have a severe impact on the final results.

Another important physical phenomenon which should be addressed is the atmosphere-induced refraction of radiation from the target ship. DMAD now uses a 4/3 earth radius model to compute refraction. This 4/3 earth approximation is commonly used in radar range calculation; it is usually considered inappropriate for IR computations. LOWTRAN6 contains a more accurate model; the LOWTRAN6 refraction routine could be extracted for insertion into DMAD.

DMAD now considers the target ship to be a single point source. Another improvement to the code would be the extension of the model to multiple point sources or to an extended source.

3. Hardcopy Graphics Output

One of the most limiting features of the present version of DMAD is that it will not produce graphical output on paper. One proposed solution to this problem entails the transfer and conversion of the program to a computer such as an IBM PC-AT which would reside in the WARLAB. Since the machine would be in a TEMPEST space it need not be TEMPEST certified itself.

Graphics support primitives already exist for the PC-AT running under the new IBM Professional FORTRAN compiler. A fine quality hardcopy graphics plotter, suitable for making the graphs needed by DMAD users, is available on the market for under $900. The software to drive the plotter is also readily available.

Although the last half of DMAD would require rewriting again, the savings in user time would be substantial. Running DMAD on a dedicated machine would reduce the impact currently felt when the WARLAB computer is reserved for wargaming, its principal use.

There are several unknowns concerning the conversion of DMAD for use on a microcomputer. First, there is no guarantee that DMAD will run within the memory constraints imposed by a PC-AT (approximately 600K bytes). Dividing the program into many separately compiled load modules will almost certainly solve this problem. Secondly, it is uncertain that authorization could be obtained to place a new computer in an already crowded WARLAB environment.

4. PRINT Routine Output

When the user selects the PRINT command option, the data is presented on the video screen. Most users will prefer to have a hard copy of this data for each run of interest. Not all parameters used in the probability model are listed by the PRINT routine; many are, however. A copy of the PRINT output, handwritten notes of other variables changed using the run-time variable change routine suggested above and hardcopies of the graphics produced by the other command options would constitute a complete record of an individual program run. These would be suitable for submission to the sponsor (NAVELEX) or for submission as the solution to a class homework problem.

APPENDIX A

TRANSFERRING LOWTRAN6 TO VM/CMS

This appendix outlines the steps required to transfer the programs and data from a magnetic tape to a user's virtual machine within the VM/CMS operating system environment. The steps outlined below may vary with time; if problems develop with these procedures, consult the indicated references. Mar [Ref. 23] gives valuable information concerning the use of tapes at the Naval Postgraduate School. This manual and its predecessors were used heavily in the preparation of this appendix. Some familiarity with VM/CMS and its editor XEDIT [Ref. 24, 25] is assumed.

The first step to this transfer is to install the tape into the Computer Center's tape library. Go to the counter in Ingersoll-140 and register the tape with the duty operator. Remember the name assigned to the tape; you will use this name later. When checking this tape into the library, ask the duty operator to remove the write-enable ring. This will prevent accidental erasure of the data on the tape.

Now the properties of the tape must be determined. The desirable format is nine-track, 1600 (or 6250) bits-per-inch (BPI) density, written in EBCDIC (a representation for alphanumeric characters). Other formats, however, are acceptable. NCC ships LOWTRAN6 on nine-track, 1600 BPI, unlabeled ASCII coded magnetic tape.

To proceed toward our goal we must determine the exact tape parameters; Figure 8 lists those of importance here. Magnetic tape spools frequently have several adhesive labels listing pertinent tape data. As these labels may be incorrect we assume only that the tape is nine-track and that its name is known. For the sake of this example, we will assume that the tape is actually nine-track, 1600 BPI, ASCII encoded and non-labeled.

```
Recording density (800, 1600 or 6250 BPI)
Number of data tracks (7 or 9)
Number of files
File structure
Character interchange code (ASCII or EBCDIC)
Internal labeling (standard or unlabeled)
DSNAME (Data Set NAME) of files if standard labels are used
```

Figure 8. Magnetic Tape Parameters

Figure 9 is a listing of an EXEC2 [Ref. 26, 27] file that will provide the needed information about the tape. Use XEDIT to create this file with the name TAPESCAN JOB.

```
//NACIT  JOB (0617,0312),'CODE-61CR',CLASS=E
//*MAIN ORG=NPGVM1.0617P
//  EXEC TSCAN,VOLIN=LOWX83,DCBIN='DEN=3'
//
```

Figure 9. TAPESCAN JOB

TAPESCAN JOB is a program written in Job Control Language (JCL) [Ref. 28]. The file consists of several instructions that will be executed by the batch operating system MVS (Multiple Virtual System). Each line in the file is a logical "card". The first card is the JOB card; it logs the job into the system and identifies it for accounting purposes. The first parameter is the job name; throughout this appendix all jobs will be shown with the job name NACIT. The job name may be any string of eight alphanumeric characters, the first of which must be alphabetic. It is used to label printer output and to identify the output when returned to the user's virtual reader upon job completion. The second parameter, JOB, must appear exactly as shown. The parameters in parentheses are the user number and project number of the person submitting the job for processing. These numbers are assigned by the Computer Center's User Registration and Accounting Office. The next field contains any identifying information the user may wish to provide. This ID field may contain up to twenty characters, delimited by single quotation marks. Computer Center policy recommends the insertion of the user's full name and student mail center (SMC) number (or faculty distribution code). The final parameter declares the priority of the job; CLASS=E is sufficient for this task.

The *MAIN card allows the user to track the job as it progresses through the system and ensures that it returns to the user's virtual machine upon completion. If this card were missing, the output generated by the job would be routed to the Computer Center's MVS line printer. The four digits preceding the character "P" are the user's identification number. This must match the user ID found on the JOB card.

Each JCL file discussed in this appendix will have these same two cards as the first two lines of the file.

The EXEC card directs MVS to execute a tape utility program named TSCAN. Two parameters are passed to TSCAN; the first is the name given to the tape when it is signed into the Computer Center. The second parameter is the presumed tape density. 'DEN=3' specifies 1600 BPI, 'DEN=4' is 6250 BPI. This parameter need only be your best guess;

if it is incorrect, the program will proceed while notifying you of the correct density.

The final card, '//', forms the end of the card 'deck'; it informs MVS that the JCL 'program' has been read fully.

Assuming this file (named TAPESCAN JOB) has been created using XEDIT, we may now submit it to MVS for processing. To do so, type:

SUBMIT TAPESCAN

VM/CMS will respond by informing you that the job has been submitted to MVS and logged into the job submission file on your virtual machine.

Once submitted, you may track your job's progress through the system by typing INQ [Ref. 29] followed by the job name given on the JOB card, as in:

INQ NACIT

Once the job has been submitted for processing you need not remain logged onto VM/CMS; as this job requires tape drive resources, a wait of from several minutes to an hour or more may be anticipated. When the scan has been completed, the operating system will send a message to the virtual reader of the user whose USERID is specified on the JOB and *MAIN cards. The message will be similar to:

DMTAXM104I FILE (5550) SPOOLED TO 0617P . . .

The file that has been transferred to the user's virtual reader is the output file generated by TSCAN. To read this file, type:

RLOOK

Note: RLOOK is being replaced by another virtual reader utility, RDRLIST [Ref. 30].

RLOOK permits a user to manipulate the files present in a user's virtual reader. RLOOK is a rather intricate program, not entirely suitable for novice users. For on-line documentation on RLOOK and its subcommands type:

HELP

or

HELP subcommand

To see a list of the subcommands available type:

? MENU

When RLOOK is executed, a list of files in the virtual reader will appear on the screen. If more than one file is listed, look for the most recently added file. This will be the TSCAN output file. Select this file for display by typing 'S' followed by a space and the file's four digit serial number. The file seen will resemble that shown in Figure 10. Type 'F' to browse forward in the file, 'B' to move backward. Typing:

DQ

will display the reader queue again.

Notice that TSCAN reports the tape density; this is one of the tape parameters we must know to proceed with the transfer. If the condition code, COND CODE, returned is zero, the job executed properly. If the condition code is not zero, recheck the job file for errors and resubmit the job. If this does not give proper results, see the duty Computer Center consultant.

In the example of Figure 10, the output generated by the scan of only one file has been shown; each file on the tape will have a block of statements that describe it, beginning with the RECORDS line and ending with the END-OF-FILE line. The number of records in the file is given along with the minimum, average and maximum number of characters per record. Throughout the remainder of this appendix we will assume the tape contains twelve files with records of 140 characters each.

The tape provided by NCC contains, in addition to the code for the basic program LOWTRAN6, a segmented loader map (useless at non-CDC installations), a sample input data file and the two resultant output data

```
                .
                .
                .
20:05:06  +TAPE:  ==>  ==>  LOWX83 IS 1600 BPI
                .
                .
                .
IEF142I NACIT TSCAN - STEP WAS EXECUTED - COND CODE 0000
                .
                .
                .
CONTROL STATEMENT:  DMPEND(10,0)
DMPEND  (UNIT=10,MODE=0)

UNIT: 10   DDNAME: FT10F001  VOLUME: LOWX83   DEN: 1600

RECORDS      1-8399  LENGTH      140
8399  RECORD(S) PROCESSED
0  PARITY ERROR(S)               517  FEET INPUT
0  TOTAL PARITY ERROR(S)         517  TOTAL FEET
MINIMUM:    140      AVERAGE:    140       MAXIMUM:    140
END-OF-FILE      1
                .
                .
                .
END OF RUN
                .
                .
                .
```

Figure 10. TAPESCAN OUTPUT

files (useful for initial program check-out), a plotting program (unusable at NPS because it contains machine-dependent plotting calls), a filter function program (not yet implemented at NPS) and several other input, output and intermediate data files for use with these programs.

For insurance, make a backup of the original tape. The backup tape can be either a user provided tape or one checked out from the Computer Center tape library. If the tape is provided by the user it must be logged into the tape library exactly as the LOWTRAN tape was. Computer Center tapes may be checked out for this purpose; see the duty operator for details.

Figure 11 shows a sample JCL file that will perform the backup. As before, create this file using XEDIT.

```
//NACIT  JOB (0617,0312),'CODE-61CR',CLASS=E
//*MAIN ORG=NPGVM1.0617P
//  EXEC TAPE,VOLIN=LOWX83,VOLOUT=NPS624
//SYSIN DD *
CPYEND(10,11)
/*
//
```

Figure 11. TAPETRAN JOB

The JOB and *MAIN cards appear as they did in Figure 9. In the EXEC card notice we now invoke the utility program TAPE to perform the copying. We specify the input tape using the VOLIN= parameter. In this example, the input tape name is LOWX83; this was the name given to the tape when it was checked into the Computer Center's tape library. The output tape name is NPS624; this is the name of a tape borrowed from the Center. It could very well have been a tape provided by the user. Regardless of the source of the backup tape, it *must have a write-enable ring installed.* The SYSIN card tells the MVS operating system that the next lines in the TAPETRAN file are input data for the TAPE utility program. CPYEND(10,11) tells TAPE to copy all the data from unit 10 (the tape LOWX83) to tape unit 11 (NPS624).

Submit TAPETRAN JOB to MVS by typing:

SUBMIT TAPETRAN

Again, when processing has been completed, MVS will send the results to the virtual reader of the user whose USERID appears on the JOB and *MAIN cards. View this file using the RLOOK utility. Figure 12 shows part of the output generated by this job.

In Figure 12, ellipsis have been used to show deletion of non-critical information within the example output file. As before, look for

```
        .
        .
        .
IEF142I NACIT TAPE - STEP WAS EXECUTED - COND CODE 0000
        .
        .
        .
CONTROL STATEMENT:  CPYEND(10,11)
CPYEND  (FROMUNIT=10,TOUNIT=11)

UNIT: 10    DDNAME:  FT10F001    VOLUME:  LOWX83  DEN: 1600


UNIT: 11    DDNAME:  FT11F001    VOLUME:  LOWX83  DEN: 1600

8399  RECORD(S) PROCESSED
0  PARITY ERROR(S)         517  FEET INPUT   517  FEET OUTPUT
0  TOTAL PARITY ERROR(S)   517  TOTAL FEET   517  TOTAL FEET

MINIMUM:   140      AVERAGE:  140      MAXIMUM:  140
END-OF-FILE     1
        .
        .
        .
```

Figure 12. TAPETRAN OUTPUT

the line that shows the condition code, COND CODE; it should list a condition code of 0000. One common problem occurs when attempting to write on a write protected tape. Make sure that the write-enable ring is installed.

Note the similarity between TAPETRAN OUTPUT in Figure 12 and TAPESCAN OUTPUT in Figure 10. The TAPE utility program is used in both cases to read and transfer data.

At this stage, if we were not sure of the interchange format used (ASCII or EBCDIC) we would use the TAPE utility to dump the first few records of the tape in EBCDIC. If the output makes sense, the format used is EBCDIC; if not, the tape is written in ASCII. We assume here that the tape LOWX83 is an ASCII tape. As the IBM 3033 encodes its data in EBCDIC, we must have some mechanism for converting from ASCII to EBCDIC. Fortunately, the TAPE utility can perform this conversion; Figure 13 shows the necessary JCL file.

```
//NACIT JOB (0617,0312),'CODE-61CR',CLASS=E
//*MAIN ORG=NPGVM1.0617P
//  EXEC TAPE,VOLIN=LOWX83,DCBIN='DEN=3',
//           VOLOUT=NPS635,DCBOUT='DEN=4'
//SYSIN DD *
ASCIIT(10),CPYEND(10,11)
/*
//
```

Figure 13. A_TO_E JOB

In the EXEC card, we specify the density of the input tape; we determined this value by running TSCAN. The 'DEN=3' parameter specifies 1600 BPI tape. For our working tape we will use the increased density 6250 BPI tape format. The parameter for this density is 'DEN=4'. As before, the card following the SYSIN card contains input data for the TAPE utility program. The ASCIIT(10) parameter turns on the ASCII to EBCDIC conversion for the input tape. CPYEND(10,11) copies the contents of LOWX83 onto tape NPS635, converting the data from ASCII to EBCDIC as it goes.

Submit A_TO_E JOB in the same manner as the previous jobs; the output file generated will look very similar to Figure 12.

We now have a backup ASCII tape on NPS624 and an EBCDIC copy of the tape on NPS635. The process of transferring the data from tape to VM/CMS may now begin.

We use another IBM tape utility, IEBGENER [Ref. 31], to perform the data transfer. Figure 14 shows the JCL file used. The JOB and *MAIN cards are identical to those used in previous examples. The EXEC card invokes the utility program IEBGENER.

The SYSUT1 cards specify the parameters of the source tape. UNIT=3400-5 is a nine-track, 6250 BPI tape drive; the source tape is the EBCDIC coded tape NPS635. The LABEL card contains three parameters which must be specified; the first is a sequence number of the file on the tape, the second is 'NL' to denote a non-labeled tape and the third is 'IN', identifying the tape as an input tape. Note that for brevity, ellipsis have been used in Figure 14. Each file on the input tape requires a card packet; the packet begins with an EXEC card and ends with a SYSIN card. For our example tape we would have twelve such packets. The LABEL card for the second file would look like:

```
//  LABEL=(02,NL,,IN),
```

The SYSUT2 cards specify the destination for the data. We cannot send data directly from MVS to a VM/CMS virtual machine; the data must first go to a disk volume. MVS004 is a special MVS disk used for this purpose. It is an IBM 3350 disk unit; hence the UNIT=3350 parameter. The DSN (Data Set Name) is the name of the file generated by each file transfer step. Note that for the first file the DSN is 'F0617.L001'. The first letter ('F', in this case) specifies the class of

```
//NACIT JOB (0617,0312),'CODE-61CR',CLASS=E
//*MAIN ORG=NPGVM1.0617P
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-5,VOL=SER=NPS635,DISP=(OLD,PASS),
//  LABEL=(01,NL,,IN),
//  DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400,DEN=4)
//SYSUT2 DD UNIT=3350,VOL=SER=MVS004,DISP=(NEW,KEEP),
//  DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400),
//  SPACE=(CYL,(1,1),RLSE),DSN='F0617.L001'
//SYSIN DD DUMMY
         .
         .
         .
//  EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=3400-5,VOL=SER=NPS635,DISP=(OLD,PASS),
//  LABEL=(12,NL,,IN),
//  DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400,DEN=4)
//SYSUT2 DD UNIT=3350,VOL=SER=MVS004,DISP=(NEW,KEEP),
//  DCB=(RECFM=FB,LRECL=140,BLKSIZE=1400),
//  SPACE=(CYL,(1,1),RLSE),DSN='F0617.L012'
//SYSIN DD DUMMY
//
```

Figure 14. COPYLOWT JOB

user ('F' = Faculty). The next four digits are the USERID. A period, '.', separates the USERID from the name of the file. For this example, the twelve files are numbered from L001 to L012. The DSN for file number ten would be DSN='F0617.L010'.

This JCL file is submitted in the manner described above; similarly, the output is read using RLOOK. Figure 15 shows a sample output file generated by COPYLOWT JOB. There will be a 'STEP WAS EXECUTED' line for each of the twelve files transferred to MVS004. The condition codes, COND CODE, should all be 0000.

All twelve files from the LOWX83 tape have been converted to EBCDIC and transferred to MVS004. The task now is to transfer the files from MVS004 to a VM/CMS virtual reader.

```
    .
    .
    .
IEF142I NACIT - STEP WAS EXECUTED - COND CODE 0000
    .
IEF142I NACIT - STEP WAS EXECUTED - COND CODE 0000
    .
IEF142I NACIT - STEP WAS EXECUTED - COND CODE 0000
    .
DATA SET UTILITY - GENERATE
    .
PROCESSING ENDED AT EOD
    .
    .
    .
```

Figure 15. COPYLOWT OUTPUT

At this point we will probably require additional virtual disk storage. This storage will be used as a temporary "scratch pad" to hold the LOWTRAN6 files. Figure 16 is an EXEC2 file (TEMPSTOR EXEC) that will obtain the amount of storage needed to continue with the LOWTRAN transfer. Run this EXEC2 file by typing TEMPSTOR. TEMPSTOR EXEC reserves eight cylinders of virtual disk space; the filemode of this temporary storage is 'T'.

We use the EXEC2 program GETMVS to transfer the programs and data from MVS004 to the virtual disk. Type GETMVS to run this program.

You may obtain a list of all files on MVS004 by responding to the first question with a 'Y'; you may skip the list by typing 'N'. You already know the names of the files on MVS004 you are interested in so the list of files is unnecessary; recall that the names of the files are 'F0617.L001' through 'F0617.L012'.

```
&TRACE OFF
CLRSCRN
&TYPE OBTAINING TEMPORARY STORAGE
CP DEFINE T3350 AS 355 CYL 8
&IF &RC = 92 &GOTO -ALREADY_DEFINED
&BEGSTACK -END1
YES
TEMP
-END1
SET CMSTYPE HT
FORMAT 355 T
-ALREADY_DEFINED
ACCESS 355 T
SET CMSTYPE RT
&EXIT
```

Figure 16. TEMPSTOR EXEC

When the computer asks 'Dsn?', type the name of the first file like this:

F0617 L001

Note the decimal point has been replaced with a blank. Information concerning the space requirements of the file will now appear on the screen. Hit the 'ENTER' key to continue to the next screen.

GETMVS now asks for the filename, filetype and filemode you wish to assign to the new file. Type these in with spaces between them. For the first file, F0617 L001, we type:

LOWTRAN FORTRAN T

The file is transferred to the temporary disk. GETMVS allows us to browse through the file to make sure it is what is expected. If the program looks acceptable, answer 'Y' to the "Save file? (Y or N)" question.

For the purpose of this example we will transfer only the first file, LOWTRAN6. The remaining eleven files are transferred to VM/CMS in exactly the same fashion.

When asked for the next "Dsn?", press enter. LOWTRAN6 should now be present on the 'T' disk. Type:

```
FLIST * * T
```

When we used TSCAN to view the tape LOWX83 we observed that the record size was 140 characters. We see the same results in the FLIST display. FORTRAN IV files have record sizes of 80 characters; we must truncate each record in the LOWTRAN file to 80 characters. Exit FLIST and type:

```
COPY LOWTRAN FORTRAN T LOWTRAN FORTRAN T (REPLACE LRECL 80
```

XEDIT cannot manage a file the size of LOWTRAN6. To permit easy editing we break LOWTRAN6 into three smaller files. Type:

```
COPY LOWTRAN FORTRAN T LOWTRANA FORTRAN T (FR 1 FOR 3000
COPY LOWTRAN FORTRAN T LOWTRANB FORTRAN T (FR 3001 FOR 3000
COPY LOWTRAN FORTRAN T LOWTRANC FORTRAN T (FR 6001 FOR 3000
```

The remaining eleven files on MVS004 should be transferred in exactly the same manner. FORTRAN IV files should have a record length of 80; listing files should have a record length of 133. Remember that the 'T' disk is temporary. Move those files to be kept to the 'A' disk, as space allows.

APPENDIX B

CONVERTING LOWTRAN6 TO IBM FORTRAN IV

We assume here that the procedures of Appendix A have been followed and that LOWTRAN6 is now present on the temporary 'T' disk in three files: LOWTRANA, LOWTRANB and LOWTRANC. Modify these files according to the directions outlined below using the XEDIT program editor. The three files will be combined later for compilation.

The FORTRAN IV (H Extended) compiler was used for the conversion of LOWTRAN6 [Ref. 32, 33]. The steps in the conversion process are:

* CDC FORTRAN requires a PROGRAM statement not used in the standard; the statement at lines LWT 105 and LWT 110 should be removed or made a comment by placing a 'C' in column one of those cards.

* Under CMS, files are opened by issuing FILEDEF statements to the operating system. The OPEN statements at lines LWT 2830, LWT 2835 and LWT 2840 should be made comments.

* Named BLOCK DATA statements are not allowed by the IBM compilers; AFGL has provided BLOCK DATA statements both with and without names in each BLOCK DATA subprogram. The statement with the name should be made a comment and the commented unnamed statement should have the comment indicator in column one removed to make it an active statement. This change affects every BLOCK DATA subprogram in the code.

* In the subroutine FNDHMN, the value of the variable ETA in the DATA statement at line HMN 185 should be changed to 1.0E-06; the rationale behind this change will be explained later.

* Insert the statement

```
DATA EPSILN/1.0E-06/
```

immediately following statement FLL 165 in the subroutine FILL. In the same subroutine, comment the statement on card FLL 260 and insert these two statements following card FLL 260:

```
DIFF = (HB-Z(I))
IF (DIFF.GT.EPSILN) GO TO 120
```

* In the subroutine LAYER, make the statement on card LAY 340 a comment. Immediately following this card insert the statement:

```
IF ((DENA(K).LT.EPSILN).OR.(DENB(K).LT.EPSILN)) GO TO 100
```

* In the subroutine DEL, card DE 155 should be a comment because the variable EPSILN is never referenced.

* In the subroutine SUBSOL, card SBS 245 should be replaced by the two cards:

```
     IF (NDAY(I).GT.IDAY) GO TO 20
 10  CONTINUE
```

This substitution is suggested because the standard requires that the last statement of a DO loop construct not be a GO TO statement of any form including that contained in a logical IF statement.

* The subroutine CIRRUS issues a call to a subroutine RANSET to initialize a random number generator. The random number generator used in the AFGL version of LOWTRAN6, RANF, is a CDC product and hence unavailable at IBM installations. In the NPS version, RANF has been replaced by the NPS NONIMSL routine RN, which has no

initialization function. Consequently, make card CIR 505 a comment and replace card CIR 510 with:

```
 15    URN = RANDOM(IDUM)
```

* In the function RANDOM, make card RDM 115 a comment and insert, immediately following it, the card:

```
       RANDOM = RN(0)
```

Any uniformly distributed random number generator could be used as a suitable replacement for RANF.

This concludes the list of changes to be made to the AFGL LOWTRAN6 to permit compilation by an IBM FORTRAN IV compiler. As evident through analysis of the changes, several were not merely syntactic in nature. Changes in program logic were mandated largely by a difference in how the two computers involved (the CDC 6600 and the IBM 3033AP) represent floating point (real) numbers.

The basic word length for the CDC 6000 series computer is 60 bits; the word length for the IBM 3033 is 32 bits. This difference proves to be the most difficult obstacle to overcome when undertaking a conversion of this nature. This problem manifests itself in two major areas. First, the number of characters that can be stored in a single word on the CDC machines is ten (six bits per character) while the capacity of the IBM word is four (eight bits per character). AFGL carefully avoided placing more than four characters in each real variable. The second word length problem is far more serious; it concerns the number of significant digits carried for each real number.

The CDC single precision REAL variable has better than fourteen significant decimal digits, nearly the precision of the IBM DOUBLE

PRECISION (REAL*8) variable. Also, the CDC real number has more bits devoted to the exponent than either the IBM REAL*4 or REAL*8 variables. The exponent problem is not addressed in the NPS implementation; the magnitude range of REAL*4 and REAL*8 variables is of order 10^{-78} through 10^{+75}, sufficient for the purposes of this program.

Two methods exist for solving the mantissa size problem. The REAL*4 variable possesses a precision of approximately six decimal digits; the REAL*8 variables, approximately fifteen. To promote manually each real variable in the code to the next higher class, i. e. from single precision to REAL*8, is too costly in terms of programmer time and risk as inadvertent type conversion and memory sharing problems (through EQUIVALENCE statements) may lead to unpredictable results.

The solution selected at NPS was to utilize the automatic precision increase facility of the IBM FORTRAN IV (H Extended) compiler. Invoking the compiler with the parameter 'AUTODBL (DBLPAD)' forces the promotion and padding of single and double precision variables throughout the program. Promotion is the process of converting real variables of one precision to the next higher precision. Padding is the process of doubling the storage size of non-promoted items; this preserves the relationship between promoted and non-promoted items sharing storage (as through EQUIVALENCE constructs).

APPENDIX C

COMPILING LOWTRAN6

Use the EXEC2 file of Figure 17 to compile the LOWTRAN6 program, converted using the methods outlined in Appendix B. Here we assume that the program remains in three files on the temporary 'T' disk;

```
&TRACE OFF
CLRSCRN
&BEGTYPE  -END1

NAVAL POSTGRADUATE SCHOOL
LOWTRAN6 MAIN PROGRAM COMPILATION
VERSION 1.0     {09NOV83}

-END1
ERASE LOWTRAN6 FORTRAN T
ERASE LOWTRAN6 LISTING T
ERASE LOWTRAN6 TEXT T
COPYFILE LOWTRANA FORTRAN T LOWTRAN6 FORTRAN T
COPYFILE LOWTRANB FORTRAN T LOWTRAN6 FORTRAN T (AP
COPYFILE LOWTRANC FORTRAN T LOWTRAN6 FORTRAN T (AP
*
*  NOTE:  THIS PROGRAM SHOULD BE COMPILED WITH THE SWITCHES
*         SET AS SHOWN BELOW.  THESE SWITCH SETTINGS ENABLE
*         THE USER TO OBTAIN RESULTS COMPARABLE WITH THOSE
*         AVAILABLE AT CDC INSTALLATIONS SUCH AS AFGL.  A
*         NOTE OF WARNING:  AUTOMATIC PRECISION INCREASE
*         DOES NOT MAKE THE MODEL BETTER OR THE RESULTS
*         MORE ACCURATE.  API IS USED TO FACILITATE THE
*         COMPARISON OF RESULTS OBTAINED USING NPS/LOWTRAN6
*         WITH THOSE THAT MAY BE OBTAINED USING AFGL/
*         LOWTRAN6.
*
FORTHX LOWTRAN6 (NOTERM NOLIST ALC AD (DBLPAD) OPT (2)
```

Figure 17. COMPLOW EXEC

LOWTRANA, LOWTRANB AND LOWTRANC. Recall that the program editor XEDIT cannot read a file the size of LOWTRAN6; the solution is to break the file into three smaller files and reconstruct them at compile-time.

Once this EXEC2 file has been entered, it may be invoked simply by typing its name, 'COMPLOW'.

The erase instructions are present to remove any files that may be present from previous attempts at compiling the code. The COPYFILE instructions assume that the three pieces of LOWTRAN6 are present on the temporary disk volume 'T' set up in Appendix A.

When the program has been compiled and found to generate correct output, the object file (LOWTRAN6 TEXT) should be transferred to a permanent disk volume. We assume here that the disk chosen is the user's 'A' disk.

The IBM FORTRAN IV (H Extended) compiler is a software product licensed from IBM. Its use will be discontinued, at NPS, when the IBM VS FORTRAN compiler incorporates the precision increase feature. When this occurs, the EXEC2 program in Figure 17 will require modification to take into account this new compiler and its switch settings.

APPENDIX D

RUNNING LOWINPUT

LOWINPUT may be run using the EXEC2 file of Figure 18.

```
&TRACE OFF
CLRSCRN
&BEGTYPE  -END1

NAVAL POSTGRADUATE SCHOOL
LOWTRAN6 INPUT DATA FORMATTING PROGRAM
VERSION 1.0     {09NOV83}

-END1
GETFMADR
&READ VARS &STAR &MODE1 &CUU
CP LINK 0029P 191 &CUU RR
ACC &CUU &MODE1
GLOBAL TXTLIB FORTMOD2 MOD2EEH NONIMSL IMSLSP EUDSTXT
FILEDEF 07 DISK LOWTINP DATA A (RECFM FB LRECL 80 BLKSIZE 80
LOAD LOWINPUT (START CLEAR
REL &MODE1 (DET
```

Figure 18. LOWINPUT EXEC

The output file generated by LOWINPUT (LOWTINP DATA) remains on the user's 'A' disk for later processing by LOWTRAN6.

APPENDIX E

RUNNING LOWTRAN6

Figure 19 contains the EXEC2 file that will run the LOWTRAN6 file compiled in Appendix C (LOWTRAN6 TEXT).

```
&TRACE OFF
CLRSCRN
&BEGTYPE  -END1

NAVAL POSTGRADUATE SCHOOL
LOWTRAN6 ATMOSPHERIC PROPAGATION MODEL
VERSION 2.0     {22AUG84}

-END1
GLOBAL TXTLIB FORTMOD2 MOD2EEH IMSLSP NONIMSL
FILEDEF 05 DISK LOWTINP DATA * (RECFM FB LRECL 80 BLKSIZE 80
FILEDEF 06 DISK LOWTOUT LISTING A (RECFM FB LRECL 132 BLKSIZE 132
FILEDEF 07 DISK LOWPLTIN DATA A (RECFM FB LRECL 132 BLKSIZE 132
LOAD LOWTRAN6 (START CLEAR
&BEGTYPE  -END2

EXECUTION COMPLETE.
-END2
```

Figure 19. LOWTRAN EXEC

The output file generated by LOWTRAN6 is called LOWTOUT; if execution of the program proceeds normally, it will be found on the user's 'A' disk. The data is written to a file instead of writing it directly to a line printer in the interest of saving paper. The user may read the output (using the VM/CMS utility program BROWSE) before deciding upon its ultimate destination. To erase the file, type:

ERASE LOWTOUT LISTING A

To print the data, type:

PRINT LOWTOUT LISTING A

A file, LOWPLTIN, used as input to the plotting program LOWPLOT is also generated by LOWTRAN6 and will be found on the 'A' disk.

APPENDIX F

NPS DISPLAY MANAGEMENT ROUTINES

Three programs, RELEAS, EUD001 and EUD002, have been written by programmers at the Naval Postgraduate School Computer Center to make usage of the IBM utility package 'DMS/CMS' easier. DMS permits the definition of interactive screen panels that may be displayed by a program. A panel may display information and have areas for user input. This system makes the process of obtaining input for a program much more attractive and understandable for the user.

DMS permits a maximum of ten active panels; RELEAS (Figure 20) releases all panels active at the time of the call, allowing more panels to be accessed. RELEAS should be called periodically in a program using DMS to ensure the maximum of ten active panels is not exceeded.

```
RELEAS   CSECT
         SAVE     (14,12),,*
         REGEQU
         BALR     R12,0
         USING    *,R12
         LR       R3,R13
         LA       R13,SAVEAREA
         ST       R13,8(0,R3)
         ST       R3,4(R13)
         B        BEGIN
SAVEAREA DS       18F
BEGIN    DS       0H
         RELEASE
         L        R13,SAVEAREA+4
         RETURN   (14,12),RC=0
         END
```

Figure 20. RELEAS ASSEMBLE

EUD001 is a function which returns the address of the variable passed as the argument to the function. EUD002 is a subroutine which displays the panel and loads and unloads the variable fields contained in the panel. Listings of EUD001 and EUD002 may be found in the NPS Technical Note (VM-09) on the subject of DMS.

APPENDIX G

THE LOWPLOT PROGRAM

For ease of operation, an EXEC2 file named LOWPLOT is used to run the FORTRAN program. The LOWPLOT exec is a modified copy of the NPS Computer Center execs, DISPLA and DISSPLA; invoking the LOWPLOT exec causes necessary disk linkages and FILEDEFs to be issued to VM/CMS. The LOWPLOT exec then directs the program loader to load and execute the main program, LOWPLOT.

The data file LOWPLTIN DATA contains the results computed as a result of a single LOWTRAN6 run. This file consists of one or more output data sets; each complete set of parameters input to LOWTRAN6 generates a single output data set. LOWPLOT reads the file and displays the first data set. The user may plot this data set by pressing PF key 12 (or 24). To view the next data set in the file without plotting the current set hit the ENTER key. PF key 03 (or 15) halts program execution and returns the user to VM/CMS.

Once the desired data set has been selected (by pressing PF key 12/24), LOWPLOT begins prompting the user for the plot parameters. The vital plot parameters include the horizontal and vertical page lengths, the selection of an optional page border, and horizontal and vertical graph dimensions.

With one exception (to be covered later), LOWPLOT computes all distances in centimeters.

Reading the data set, LOWPLOT determines whether the data set is the result of a transmittance or a radiance calculation. If a transmittance calculation has produced the data, the user is asked to specify the x-axis units (micrometers or cm^{-1}) and the type of plot desired (both axes linear, x-axis linear and y-axis logarithmic, or x-axis logarithmic and y-axis linear).

For radiance data, the user specifies the desired plot type, and if the data to be plotted is the result of emitted radiance, scattered radiance, ground reflected radiance or total radiance calculations.

In transmittance mode, the following results may be plotted: total, H_2O, CO_2, O_2, N_2, H_2O vapor, molecular, aerosol and HNO_3 transmittances and aerosol and integrated absorptions. For both radiance and transmittance modes the user is then shown the initial, final and increment values for the frequency and wavelength of the data; the user provides values necessary to specify the axes.

An interpolation method known as the Rational Spline Method is used for data interpolation and smoothing. The user is directed to the online program documentation and to the DISSPLA User's Manual for a complete description of this method.

Optional grid lines may be drawn; these are particularly useful on logarithmic plots. Furthermore, the graph may be framed with an optional, variable thickness border.

The graph is then drawn on the Tektronix 618 video graphics display terminal. If any errors occurred in the plotting routine or if values in the data set do not fall within the user supplied graph limits, advisory error messages are displayed on the 3277 terminal screen. To take action on any error messages, the user should reselect the data set and begin the process of plotting again. The screen erase key is used to clear the screen of any messages that may appear. In many cases, the graph will not appear on the 618 screen until all error messages have been acknowledged with the erase key.

If a hard-copy is desired, the 'hard-copy' button on the front of the 618 should be depressed, directing the image to the hard-copy device attached to the terminal. The location of the hard-copy unit is usually indicated clearly on the front of the 3277/618 terminal set.

When the user is finished with the currently displayed graph and desires to plot another or to exit the program, depressing the ENTER key should cause the 618 screen to be erased and the next data set in the file LOWPLTIN to be displayed on the 3277 terminal. Note that it is the next data set and not the previously plotted data set that is displayed.

APPENDIX H

THE NPS DMAD CONVERSION

A. CODE CONVERSION

The code conversion process consisted of two steps. In step one, syntactical differences between the FORTRAN compiler used at NRL and that used at NPS were resolved. The second step was concerned with the extensive rewriting of the many graphical subroutines which comprise the latter half of the program.

The VAX FORTRAN compiler [Ref. 34, 35] was used in the NPS implementation of DMAD. The DI3000 graphics package [Ref. 36] was used to produce the large quantity of useful graphics output available from the NPS DMAD program.

None of the mathematical logic, i. e. no code before the routine EXECGR, required modification as a result of the move.

1. Syntactical Differences

(1) VAX FORTRAN requires an exclamation mark '!' to delimit comments on code lines; DMAD originally used the '/*' combination.

(2) The 'INCLUDE' statement in VAX FORTRAN must begin in column six, as any other statement. PRIME 'INCLUDE' statements begin in column one.

(3) The mechanisms of opening and closing files differ between the two systems; these differences must be resolved to permit the use of data files.

2. Differences in Graphics and Support Subroutines

The latter half of the DMAD code consists of graphics and support subroutines. These routines tended to be very dependent upon the PRIME minicomputer system used at NRL. Consequently, they required extensive rewriting so that DMAD could be run on the NPS WARLAB VAX-11/780. Because of the significant alterations to the code, the function of each routine is discussed rather than describing the modifications made to the NRL code.

If this program were to be converted to another computer, the programmer need only duplicate the functions performed by each routine.

Those routines affected by the NPS implementation are:

(1) EXECGR. Permits the user to select from one of fifteen program options, including plotting graphs, reading and writing effectiveness data files, listing help information and printing important run parameters. EXECGR calls the following routines: HELP, NL$IO, PPLOT, PPLOT2, PPLOT3, PRINT, RAYPTH, RCT2, RCTPLT, RDEFF, RESULT, VPLOT, WANT AND WREFF.

(2) WANT. Passed a character string, containing a yes-or-no response question, WANT prints the string on the terminal and elicits a reply. The function returns a logical (TRUE/FALSE) value.

(3) GNPLTB. Called by RAYPTH to initialize a graph, draw axes and display each of the six data sets comprising plot P9. GNPLTB calls BGLIN$ and PLNUMB.

(4) BGLIN$. GNPLTB passes two singly dimensioned arrays to BGLIN$; the data in these arrays is plotted to the RAMTEK display screen.

(5) NICESC. This routine scales data to provide "nice" graphical bounds and major tickmark intervals. It remains unchanged from the original version of DMAD.

(6) HELP. When called by EXECGR, HELP prints, to the monitor, a brief description of each command option.

(7) NICNUM. NICNUM returns the real representation of a "nice" integer just greater than the value passed to it. The following numbers are considered "nice": -10.0, -5.0, -2.0, -1.0, 0.0, 1.0, 2.0, 5.0 and 10.0. The input parameter is tested against each of these numbers to determine the correct output value. In the NRL DMAD code, test values such as 5.0 were used. Because of differences in the internal numeric format between the two computers, test values such as 5.00001 are now used. This reduces the impact caused by inherently inaccurate floating-point arithmetic.

For example, if NICNUM is passed 5.0000001292, a value of 5.0 (as 5.0000001292 is less than 5.00001) is returned. Similarly, if NICNUM receives 4.9999997, the routine returns 5.0. If the value 5.1 is passed to NICNUM, a value of 10.0 is returned.

(8) PGRID. This routine draws a polar plotting grid, complete with azimuth and range labels. PGRID calls PLNUMB to label each graph.

(9) PLNUMB. PLNUMB is used in the construction of all nine plots. The subroutine writes the NACIT title, the name of the plot, the current date and the name of the effectiveness file currently in use. If

the plot is a sector run, the number of azimuths selected and their limits are written to the plot.

(10) POLAR. PPLOT calls POLAR, passing two one-dimensional arrays of plot data. Following a call to PGRID to set up the plot, POLAR draws the graph. POLAR also calls MNMX and NICESC.

(11) PINITT. PINITT initializes many plotting variables. Most of these variables are vestigial; as time permits, this routine should be pared down with an eye toward its ultimate elimination.

(12) PPLOT. Called by EXECGR, PPLOT is responsible for plotting graphs P2, P3 and P4. PPLOT calls POLAR to perform the plotting; labels are written by PPLOT.

(13) PPLOT2. Also called by EXECGR, PPLOT2 draws plot P5. After determining the effectiveness threshold to be used in the plot, PPLOT2 calls PGRID. PPLOT2 plots the locus of points for which the effectiveness exceeds a user defined value. This routine also calls WANT.

(14) PPLOT3. EXECGR calls PPLOT3 to plot graph P6. Initially, stored thresholds are used to make this graph of the points where the effectiveness crosses given threshold values. On subsequent P6 plots, the user is asked for the number and value of the thresholds to be considered. PPLOT3 calls PGRID for plot initialization. The subroutine WANT is also called.

(15) PRINT. PRINT remains essentially unchanged. This routine should be modified to permit the writing of data to a file for later hard-copy generation on a line printer.

(16) RAYPTH. RAYPTH, through a call to GNPLTB, makes plot P9. Optical paths for each scanner look-down angle are drawn. A 4/3 earth refraction correction plot is also written.

(17) RCT2. Plot P7 is drawn by RCT2. The following routines are called: ANORM, PLNUMB, TICVAL and WANT.

(18) ANORM. Given an arbitrary angle, ANORM normalizes it to a value within the range [0,360).

(19) RCTPLT. Plot P1 is generated. PLNUMB, TICVAL and WANT are called.

(20) RDEFF. RDEFF reads a previously written effectiveness file. Effectiveness files are used so the user is not required to make a full probability model run just to review data; the results of a run may be saved for later analysis. RDEFF is called by EXECGR.

(21) SAVRES. Called by RDEFF and WREFF, SAVRES saves and restores effectiveness data files.

(22) TICVAL. TICVAL computes major graph tickmark values and separations.

(23) VPLOT. This routine draws plot P8. It is called by EXECGR and in turn calls BACKGD, DD963, GETMAX, LINER, PROSE and WARROW.

(24) GETMAX. GETMAX considers all X and Y values from the target ship's outline, its position at decoy launch, the decoy deployed positions and the decoy final positions. It returns the maximum values of X and Y found. GETMAX is essentially a scaling routine used by VPLOT to determine proper plot size.

(25) BACKGD. BACKGD plots axes and tickmarks for plot P8. The routine is called by VPLOT.

(26) WARROW. This routine plots a wind vector on plot P8. WARROW is called by VPLOT. It calls LINER and SKIPR.

(27) SKIPR. SKIPR performs a relative move using polar coordinates.

(28) LINER. LINER performs a relative draw in polar coordinates.

(29) PROSE. PROSE writes text information concerning plot P8 to the RAMTEK monitor.

(30) DD963. The outline of a DD963 class destroyer is drawn on plot P8 by the subroutine DD963.

(31) WREFF. WREFF writes an effectiveness file to disk.

(32) CLRSCN. The terminal monitor is cleared.

(33) SETUP. This routine asks the user to supply information about the RAMTEK monitors to be used for a specific run. It then issues initialization instructions to the RAMTEK hardware to prepare them for plotting DMAD graphs.

(34) MNMX. The minimum and maximum values in an array passed as parameters to MNMX are returned.

B. COMPILATION AND LINKAGE EDITING

Code compilation is simple but time consuming. To compile DMAD, type:

```
FOR/CONTINUATIONS=30 DMAD
```

The continuations parameter is required because the number of contiguous continuation lines in the code exceeds the default acceptable value for the compiler. The compiler will generate a listing file (DMAD.LIS) showing source code and a cross reference listing. The compiler also generates an object file (DMAD.OBJ). This object file must be linked to various run-time libraries to permit usage of scientific and DI-3000 graphics subroutines. To perform this linkage, type:

```
SLINK DMAD
```

Don't be alarmed if several compilation warning error messages appear. The file DMAD.EXE should now exist in the directory; it is now ready to run.

APPENDIX I

DMAD OPERATING INSTRUCTIONS

DMAD is simple to use; here we assume the user has access to and has an account on the NPS WARLAB VAX-11/780 computer.

To use DMAD one must first log into the system. Sit down in front of a terminal and turn the terminal on; the power switch is on the rear left side of the device. Following a short warm-up period you should see a cursor in the upper left corner of the screen. Press the return key once. VAX/VMS should respond with:

ENTER USERNAME:

Respond by typing the USERNAME for the account that contains the DMAD program and its data files. Remember to hit the RETURN key after typing in the USERNAME. The system will answer:

ENTER PASSWORD:

Type the password (the password will not appear on the screen). Again, press the RETURN key when you have finished typing the password. The system will usually now print general user login information.

If you have made an error in either the USERNAME or PASSWORD fields, the system will respond by typing:

USER AUTHORIZATION ERROR

If this happens, try typing the USERNAME and PASSWORD data again. If this condition persists after trying carefully several times, see the system operator to verify that the account exists.

When you have successfully logged into the system, you should see a '$' prompt. Type:

ASSIGN SYS$INPUT FOR001

This command tells the operating system how to handle input and output to and from DMAD. Don't forget to press the RETURN key. Now you are ready to run the program. Type:

RUN DMAD

The operating system will now load and execute DMAD. The program asks:

Enter the number of monitor sets to use: (1-3):

The monitors in the WARLAB are wired in pairs. There are presently three pairs of RAMTEK monitors; type in the number of sets you want to use in the graphics portions of the program. If you are giving a presentation to a large group you may wish to use more than one monitor pair. If you answer this question by typing the numeral 1, the program will ask you:

Enter the number of the monitor set: (1-3):

Each monitor pair is assigned a number; that number is found written on a paper sticker on the monitor screen bezel. Type that number. If you have specified the use of more than one monitor pair, DMAD will ask you for the numbers of the sets you wish to use. Enter your data as before, followed by a RETURN.

The program now asks:

BYPASS PROBABILITY EFFECTIVENESS MODEL?

If you have already run the program for the parameter values of interest and the results are on file, answer with an 'Y' (and RETURN).

This answer permits you to bypass the long, and in this case unnecessary, effectiveness computation step. This is referred to as the 'graphics only path'.

If you wish to compute the effectiveness for a new set of data, answer with a 'N'. The calculation will take a while to complete; when it has finished, the effectiveness table will be displayed on the terminal screen. This is termed the 'normal path'.

The program will then ask you to:

```
SELECT COMMAND:
P1,P2,P3,P4,P5,P6,P7,P8,P9,
PRINT,NL,WRITE,READ,EXIT,HELP
```

You may now respond with the name of the command option desired. When execution is complete, respond by typing EXIT.

APPENDIX J

DMAD COMMAND OPTIONS

Fifteen DMAD commands are available to the user. To invoke a command, type its name when DMAD directs:

```
SELECT COMMAND:
P1,P2,P3,P4,P5,P6,P7,P8,P9,
PRINT,NL,WRITE,READ,EXIT,HELP
```

The commands available are:

(1) P1: Decoy acquisition probability vs. missile range at time of decoy launch. Plot P1 is a rectangular plot; it may be annotated, if desired, with labels identifying the reaction time, acquisition region and, if applicable, burnout region. The acquisition probability is averaged over all missile approach angles. There are two X axis scales: the lower scale displays range in nautical miles, the upper scale shows time to impact at the time of decoy launch (in seconds). The value of the integral figure of merit and the average effectiveness are displayed in the left margin of the plot.

(2) P2: Decoy acquisition probability vs. missile approach angle. P2 is a polar plot. The acquisition probabilities are averaged over missile ranges at the time of decoy launch.

(3) P3: Ship lock-on range plot. The value of the program variable MODER specifies the lock-on range pattern mode. If one, the pattern is a circle. Setting MODER equal to two gives an elliptical pattern. A value of three yields two intersecting ellipses.

(4) P4: Decoy acquisition probability vs. missile approach angle. When asked, type in the missile range at decoy launch, in nautical miles.

This plot displays the acquisition probability for a given missile range. You may repeat this plot for different ranges to display the relationship between acquisition probability and the time available for the missile to acquire the decoy. P4 is a polar plot.

(5) P5: Locus of points for which effectiveness is greater than some threshold. DMAD asks for an effectiveness value. P5 is a polar plot.

(6) P6: Effectiveness isograph. P6 plots, in polar form, the ranges and bearings at which the decoy effectiveness passes through the indicated threshold. The initial threshold is provided to DMAD in the routine INIT1; this value is used for the first P6 plot. DMAD will ask the user for effectiveness thresholds for subsequent plots.

(7) P7: Plot of acquisition probability for given sector(s). P7 is a rectangular plot that displays the decoy acquisition probability averaged over a user specified set of sectors. P7 differs from P1 in that P7 shows acquisition probability *averaged over specified sectors.*

(8) P8: Ship orientation, decoy position and wind vector plot. P8 is a rectangular plot showing these vector relationships.

(9) P9: Plot of geometry of optical trajectories. P9 is a plot of the raypaths formed by each of five scanner lines as they travel out from the missile. The 4/3 earth radius approximation used in microwave refraction calculations is used to compute the refraction experienced by the IR raypaths as they transit the atmosphere. Ray height is plotted against the distance from the seeker.

(10) PRINT. Important run parameters are listed on the display screen.

(11) NL. The user is allowed to alter program parameters and recalculate decoy effectiveness. This routine is presently not implemented. It should be the first addition when changes are made to DMAD.

(12) WRITE. Saves the decoy effectiveness results for later processing.

(13) READ. Reads previously calculated effectiveness results.

(14) EXIT. Stop execution of the program. Returns the user to VAX/VMS.

APPENDIX K

A LOWTRAN6 SAMPLE RUN

We assume here that the user has a VM/CMS account on the NPS IBM 3033AP computer. The steps outlined below are those required to duplicate the text example.

First, login and link to the LOWTRAN6 mini-disk. Perform the linkage by typing:

CP LINK TO 0617P 195 AS 299 RR

The password is "LOWTRAN". Access the disk by typing:

ACCESS 299 L

This will make the LOWTRAN6 disk volume your L disk. Ensure sufficient space exists on your A disk before proceeding; some intermediate and output data files will be stored there. A good guideline is to have the A disk *filled to not more* than seventy percent of capacity.

Type:

LOWINPUT

This will start the LOWINPUT program. After a brief wait, you should see the first input panel. It asks for the atmospheric model to be used in this run. Type a '3', followed by depressing the ENTER key. The next panel should now be visible. Indicate the path type (slant path between two altitudes) by typing a '2'. Note that the cursor goes to the next input area automatically. Type a '0', followed by the ENTER key to specify the calculation mode (transmittance).

Note that the ENTER key is used to tell the computer that you are finished typing data on a panel. Do not strike the ENTER key before you (or the computer) has filled all the input blocks.

The next panel is an example of one where default values are used and displayed. If, as in the case of our example, you wish to run the program with default temperature, pressure, water vapor and ozone profiles, merely press the ENTER key. The default values shown, '0', will be entered as data to LOWTRAN6.

Read the next panel carefully. Skip the first input field by hitting the large "bent-arrow" key just above the ENTER key. The cursor should now be in the second input field. Type a '1' to suppress the printing of the transmittance/radiance table and atmospheric profiles. Of course if you wish to look at these you may type a '0'. Regardless of your response to the second field you must type ENTER to go on to the next panel.

Select the Navy maritime aerosol extinction model by typing '3' (ENTER). We use the default values for the next two panels, as our path will not extend into either the tropospheric (two to ten kilometers) or stratospheric (ten to thirty kilometers) regions. The desired air mass character is that for the open ocean; when asked, type a '1' (ENTER).

We choose to use the provided defaults again in the next panel. We do not include cirrus cloud attenuation or the Army vertical structure algorithm. Also, we do not wish to input a meteorological range as the Navy maritime aerosol model will calculate it for us. The current wind speed is 5.15 m/s. Remember to use the "bent-arrow" key to move between input fields; only use the ENTER key to move to the next panel. The twenty-four hour average wind speed is 3 m/s. For our example, the precipitation rate is zero (the default).

Study the next panel carefully. Note the letters on the left side of the text. In all cases you must enter the initial altitude. Two more parameters must be specified. To know which two belong together, match up the letters. For example, the two items marked with an 'a' go together. You could specify the final altitude and the initial zenith angle. These three parameters would be sufficient to completely describe the geometry of the desired simulation.

For our example, the initial altitude is the height of the target ship (11 m, .011 km). Use the "bent-arrow" key to move to the input field for the final altitude. Type the missile flight altitude (40 m, .040 km). Again, use the arrow key to move to the path length field. The range between the missile and the ship at the instant we are interested in is 10.9 km. Hit the ENTER key to end this panel.

Hit ENTER again to specify the defaults for the next panel.

Since we are concerned with the spectral region from eight to twelve micrometers we will specify that band here. The initial frequency is 833.333 cm^{-1} (12 micrometers). The final frequency is 1250.000 cm^{-1} (8 micrometers). The frequency increment must always be a multiple of five; for the example, type '5', followed by ENTER.

The last panel permits you to either end the program, read in a new set of data for a new run, read in only the path geometry data or read in only the frequency range and increment data. For our example, type '0' (ENTER). This will cause LOWINPUT to halt execution and return you to VM/CMS.

To run the model, type:

LOWTRAN

When the program has finished, the message "Execution complete" will appear. Two output data files will now be present on the 'A' disk. LOWTOUT LISTING is the tabular output of LOWTRAN6. This program may be printed on the Computer Center's line printer by typing:

PRINT LOWTOUT LISTING

The other file, LOWPLTIN DATA, is a file containing data to be plotted using LOWPLOT.

Graphs cannot be drawn on IBM 3278 terminals. To use the LOWPLOT program we must end this terminal session and move to another type of terminal, an IBM 3277 with an attached Tektronix 618 display monitor. Once logged onto VM/CMS again, type:

DEFINE STORAGE 960K

Plotting LOWTRAN6 graphs can take up a significant amount of memory; this command ensures that enough memory is present when it is needed. Type:

I CMS

to restart your virtual machine. To begin plotting, type:

LOWPLOT

The NPS LOWTRAN6 plotting program will execute after a short delay. The short delay may prove to be rather long during peak usage hours.

The first panel you should see is a synopsis of the input data for the run of LOWTRAN6 you just completed. Since a single run of LOWTRAN6 may consist of many individual data sets, this synopsis ensures that you plot the one you are interested in. In the present example, there is only one data set. If more than one were present, however, you could page through them using the ENTER key. Each data set would be displayed in turn. To plot the data associated with the synopsis currently displayed, hit the PF12 (or PF24) key.

The page dimensions that work well on the TEK 618 terminal are 35 cm for the horizontal dimension and 25 cm for the vertical dimension. You may draw a border around the entire page as a scissor guide, for example. The actual graph area must be smaller than the specified page size. Good numbers to use for the page size are 30 cm (horizontal) by 20 cm (vertical). Remember to hit ENTER to move to the next panel.

You may type a title for the graph; it must end with a '$' character. If you forget the '$', the plot will be generated but the title will not look as you expect. For the example, type:

```
Navy Maritime Aerosol Model$
```

The calculations done by LOWTRAN6 were for transmittance. We may plot transmittance vs. wavelength (in micrometers) or transmittance vs. wavenumber (cm^{-1}). For the example, select the transmittance vs. wavelength option by typing '0'. Type another '0' to specify a linear-linear plot.

Study the next panel carefully. In this case we will plot total transmittance; we could just as easily, however, have plotted transmittance due to water vapor continuum. Type '0' (ENTER).

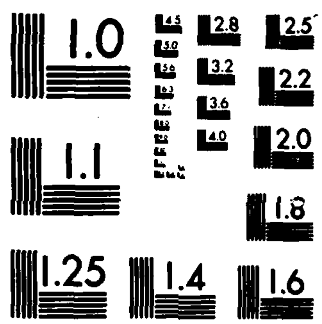
AD-A164 359 COMPUTER MODELING OF IR ENCOUNTERS AND COUNTERMEASURE EFFECTIVENESS(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA J W WILSON DEC 85 2/2

UNCLASSIFIED F/G 17/4 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

The 'X' axis origin should be 8, the step interval 0.5, and the end of the axis should be 12. The values for the 'Y' axis should be 0, 0.1 and 1.

For this plot we will disable the spline interpolation and smoothing algorithm; do this by typing '-1' (ENTER). The screen panel gives a short description of the method; consult the user's manual if additional information is desired.

Grid lines may be added; make a selection and hit the ENTER key. Similarly, the graph may be framed. Hit the ENTER key to begin the plot.

Each time a message reading "MORE..." appears in the lower right corner of the IBM 3277 screen, hit the CLEAR key; plotting will resume. When the graph is completed, the computer will ask you to press ENTER to continue. If you desire a hard-copy of the plot, press the "hard copy" button on the front of the TEK 618 monitor. Your hard-copy will be made by the device attached to the monitor.

The computer will display information about the plot on the 3277 and will eventually ask if you wish to make a VERSATEC plot of the graph. Type 'N' (ENTER).

The synopsis page should again be visible. If more than one data set is present in the LOWTRAN6 output data file, LOWPLTIN DATA, the next consecutive set, following the one just plotted, is presented. In the case of this example, there is only one data set. It is the one shown on the 3277 screen now.

To end the LOWPLOT run, hit the PF03 (or PF15) key when looking at the synopsis panel. This ends the example run of the NPS LOWTRAN6 package.

LIST OF REFERENCES

1. Shin, M-S., Calculation of Atmospheric Transmittance by IBM 3033 Computer Code LOWTRAN IIIB, M. S. Thesis, Naval Postgraduate School, Monterey, CA, 1983.

2. Air Force Geophysics Laboratory Technical Report AFGL-TR-78-0029, Atmospheric Transmission of Laser Radiation: Computer Code LASER, by R. A. McClatchey and A. P. D'Agati, 1978.

3. Guner, N., High Resolution Computer Calculation of Optical Transmittance at Sea Level Over Monterey, M. S. Thesis, Naval Postgraduate School, Monterey, CA, 1978.

4. Air Force Geophysics Laboratory Technical Report AFGL-TR-83-0187, Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 6 by F. X. Kneizys, E. P. Shettle, W. O. Gallery, J. H. Chetwynd, Jr., L. W. Abreu, J. E. A. Selby, S. A. Clough, and R. W. Fenn, 1983.

5. Air Force Geophysics Laboratory Technical Report AFGL-TR-80-0067, Atmospheric Transmittance/Radiance: Computer Code LOWTRAN 5, by F. X. Kneizys, E. P. Shettle, W. O. Gallery, J. H. Chetwynd, Jr., L. W. Abreu, J. E. A. Selby, R. W. Fenn, and R. A. McClatchey, 1980.

6. International Business Machines Corporation, IBM Virtual Machine/System Product: CMS Command and Macro Reference, SC19-6209-1, 1982.

7. International Business Machines Corporation, IBM Virtual Machine/System Product: CMS User's Guide, SC19-6210-1, 1982.

8. Naval Postgraduate School, User's Guide to VM/CMS at NPS, Technical Note VM-01, 1983.

9. Favorite, J., User's Guide to MVS at NPS, Technical Note MVS-01, 1984.

10. Bogart, J., CDC to IBM FORTRAN IV Conversion Guide, NPS Technical Note 0141-39, 1980.

11. International Business Machines Corporation, IBM Virtual Machine/System Product. Display Management System for CMS: Guide and Reference, SC24-5198-1, 1981.

12. Naval Postgraduate School, Use of the Display Management System at NPS, Technical Note VM-09, 1984.

13. Integrated Software Systems Corporation, DISSPLA Pocket Guide, 1981.

14. Integrated Software Systems Corporation, DISSPLA Users Manual, 1981.

15. Air Force Geophysics Laboratory Technical Report AFGL-TR-78-0081, FASCODE - Fast Atmospheric Signature Code (Spectral Transmittance and Radiance), by H. J. P. Smith, D. J. Dube, M. E. Gardner, S. A. Clough, F. X. Kneizys, and L. S. Rothman, 1978.

16. Clough, S. A., Kneizys, F. X., Rothman, L. S., Gallery, W. O., Atmospheric Spectral Transmittance and Radiance: FASCODE1B, SPIE Vol. 277 - Atmospheric Transmission, 1981.

17. Air Force Geophysics Laboratory Letter, Subject: Preliminary Instructions for FASCODE 1C, 14 January 1983.

18. Rothman, L. S., Gamache, R. R., Barbe, A., Goldman, A., Gillis, J. R., Brown, L. R., Toth, R. A., Flaud, J.-M., and Camy-Peyret, C., "AFGL Atmospheric Absorption Line Parameters Compilation: 1982 Edition," Applied Optics, Vol. 22, No. 15, pp. 2247-2256, 1 June 1983.

19. Rothman, L. S., Goldman, A., Gillis, J. R., Gamache, R. R., Pickett, H. M., Poynter, R. L., Husson, N., and Chedin, A., "AFGL Trace Gas Compilation: 1982 Version," Applied Optics, Vol. 22, No. 11, 1 June 1983.

20. Hudson, R. D., Jr., Infrared System Engineering, Wiley, 1969.

21. Wolfe, W. L., Zissis, G. J., (ed.), The Infrared Handbook, Environmental Research Institute of Michigan, 1978.

22. Egolfopoulos, I., Infrared Detection of Surface Vehicle: Calculation Using Atmospheric Model LOWTRAN6, M. S. Thesis, Naval Postgraduate School, Monterey, CA, 1984.

23. Mar, D. R., Magnetic Tape Usage Under MVS at NPS, NPS Technical Note MVS-07, 1984.

24. International Business Machines Corporation, IBM Virtual Machine/System Product: System Product Editor User's Guide, SC24-5220-0, 1980.

25. International Business Machines Corporation, IBM Virtual Machine/System Product: System Product Editor Command and Macro Reference, SC24-5221-1, 1982.

26. International Business Machines Corporation, IBM Virtual Machine/System Product: EXEC2 Reference, SC24-5219-0, 1980.

27. Naval Postgraduate School, Using EXEC Files Under CMS, Technical Note VM-04, 1984.

28. Brown, G. D., System 370 Job Control Language, Wiley, 1977.

29. Favorite, J., User's Guide to MVS at NPS, NPS Technical Note MVS-01, 1984.

30. Naval Postgraduate School, New Commands of VM/SP Release 2, Technical Memorandum, 1984.

31. International Business Machines Corporation, OS/VS2 MVS Utilities Manual, GC26-3902, 1977.

32. International Business Machines Corporation, IBM OS FORTRAN IV (H Extended) Compiler Programmer's Guide, SC28-6852-2.

33. International Business Machines Corporation, IBM System/360 and System/370 FORTRAN IV Language, GC28-6515-10.

34. Digital Equipment Corporation, VAX-11 FORTRAN Language Reference Manual, AA-D034C-TE, 1982.

35. Digital Equipment Corporation, VAX-11 FORTRAN User's Guide, AA-D035C-TE, 1982.

36. Precision Visuals, Inc., DI-3000 User's Guide, 1982.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Dr. A. W. Cooper Code 61Cr Naval Postgraduate School Monterey, CA 93943	2
4.	Dr. E. C. Crittenden, Jr. Code 61Ct Naval Postgraduate School Monterey, CA 93943	1
5.	Chairman, Dept. of Physics Code 61 Naval Postgraduate School Monterey, CA 93943	2
6.	LT Jeffery W. Wilson, USN NAVPRO POMONA, Code PC 1675 West Mission Blvd P. O. Box 2505 Pomona, CA 91769	2

END

FILMED

3-86

DTIC