Project Report
ATC-151

Air-to-Air Visual Acquisition Handbook

J. W. Andrews

27 November 1991

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Federal Aviation Administration,
Washington, D.C. 20591

TECHNICAL REPORT STANDARD TITLE PAGE

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
ATC-151	DOT/FAA/PM-87/30	

4. Title and Subtitle	5. Report Date
Air-to-Air Visual Acquisition Handbook	27 November 1991
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
J.W. Andrews	ATC-151
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Lincoln Laboratory, MIT P.O. Box 73 Lexington, MA 02173-9108	11. Contract or Grant No. DTFA-01-85-Z-02015
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, DC 20591	Project Report
	14. Sponsoring Agency Code

15. Supplementary Notes

The work reported in this document was performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology, under Air Force Contract F19628-90-C-0002.

16. Abstract

The document describes a set of computer programs that provide a practical means for predicting air-to-air visual acquisition performance for aircraft on collision courses. The programs are based upon a mathematical model of pilot visual acquisition performance. Guidelines are provided for selecting model parameters based upon previously collected flight test data. Selected results of computer analysis are provided.

17. Key Words	18. Distribution Statement
flight test, traffic alert, visual acquisition, visual detection, visual range	This document is available to the public through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	84	

FORM DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

This handbook provides a practical means for the prediction of pilot air-to-air visual acquisition performance. The predictions are based upon a mathematical model of visual acquisition that has been calibrated using data from actual flight tests. In order to apply the model to a particular situation, the user must understand the model well enough to 1) verify that the model is applicable to the situation of interest, 2) define the visual search conditions for the situation of interest, and 3) correctly interpret the results.

This handbook contains several tables that illustrate model results. But these pre-calculated tables cannot include all search conditions that might be of interest. For analysis of cases not provided in this handbook, one may use the computer programs (SEE1 and SEE2) that accompany the handbook. The SEE programs are interactive programs that provide visual acquisition predictions according to user inputs. They are written in the Pascal computer language and can be run on most microcomputers. SEE1 provides a basic calculation of acquisition probability. SEE2 provides a way of calculating aircraft speed limits that maintain a baseline level of visual acquisition performance. SEE1 and SEE2 share many basic computational modules.

In order to carry out the calculations, the SEE programs make certain assumptions that are not part of the model itself (e.g., SEE programs assume that the traffic is on a collision course whereas the model can be applied to any flight path). An effort has been made to keep the programs reasonably simple while preserving enough flexibility to analyze the cases of greatest interest with regard to air safety.

2. DETERMINING APPLICABILITY OF THE MODEL

Before using the model to analyze a particular case, it should be determined that the model is applicable to that case. This section discusses criteria that should be applied in order to establish applicability.

1) Search Should Occur Under Daylight Conditions

The model applies only when the aircraft is seen in bright daylight conditions (so that it is the physical structure of the aircraft that must be acquired, not the aircraft lights). The model has not been applied to the detection of strobe lights, contrails, or smoke.

2) A Value for β Must be Available

In order to apply the model, a search effectiveness parameter, β, must be selected. β is the rate of visual acquisition per solid angle of target size per second of search. It serves as as a curve-fitting parameter that accounts for all the human performance effects upon search effectiveness. β can be expected to vary with workload, pilot visual acuity, pilot training, and the amount of effort devoted to visual search. The model obtains an appropriate value of β directly from flight test data. However, flight test data is available for only a limited number of situations. If the search conditions of interest do not seem to correspond well with any documented flight test, then the only way to select β is by extrapolation. The selection of β is discussed in Section 6.1.

3) Unusual and Non-standard Visual Conditions Must be Excluded

The model does not provide any explicit adjustments for phenomena such as empty field myopia, glint, hypoxia, monocular viewing, etc. Any such phenomena present in the flight tests will be reflected in the value of β that is produced. However, if such a phenomena is absent in the flight test data, then the model provides no defined way of taking it into account.

4) Aircraft Must Approach on Non-accelerated Collision Courses

Although the general model can be applied to arbitrary flight paths, the SEE programs assume that the target aircraft is on a collision course. This is the situation of greatest interest in evaluating the reliability of visual separation. The calculations are also restricted to rectilinear (unaccelerated) flight paths. For such cases, the range (separation between aircraft) decreases at a constant rate to zero. This establishes a one-to-one correspondence between time-to-collision and range. Furthermore, the approach bearing and the visual area presented by each aircraft remain constant.

5) The Bank Angle and Pitch Angle of the Target Aircraft Must be Small

This is because the module for computing the visual size of the target assumes that the aircraft is seen from the zero-elevation plane. Hence, the upper surface of the wings and fuselage is not included in the calculation of visual area. If the bank angle or pitch of the target aircraft is not zero, the visual area of the target will be underestimated. See Section 4.2.3 for a description of the visual area calculation.

6) Cases for which the Aircraft Remain Near the Visual Resolution Threshold for Long Periods of Time Must be Excluded

The model employs a sharp cut-off in search effectiveness when target size is below a specified resolution limit (nominal value: 1 minute of arc in diameter). If the size is slightly below the resolution cut-off assumed in the model, then the model will predict that the target will never be acquired, no matter how long the pilot searches. If the size is slightly above the resolution cut-off, then the model will predict that, given enough time, the target is certain to be acquired. These all-or-nothing results may fail to reflect actual experience since a single threshold does not apply to all pilots and the threshold may actually vary with time. Fortunately, this limitation is almost never a practical concern for aircraft on collision courses since the visual target increases steadily in size and is normally well above the resolution limit before visual acquisition is likely.

7) The Model Should Not Be Used to Resolve Field-of-View Issues

The SEE program provides a simple test for whether or not the approach bearing of the target is within the pilot's field-of-view (FOV). The FOV test is described in detail in Section 4.2.4. It is intended only to provide a rough indication of the geometries for which cockpit visibility limitations will prevent visual acquisition. It does not model visibility limitations in elevation, nor does it consider the placement of window posts or other possible obstructions. Thus, it will not detect FOV limitations such as might arise when one aircraft approaches the other from far above or far below the horizontal plane. The existing FOV test is useful in providing a general correction for the effects of cockpit visibility when a variety of speeds and geometries are averaged. If it is necessary to apply the model to a specific situation where cockpit visibility is an issue, the exact field-of-view should be determined using other techniques.

8) Conclusions Must Not Be Based Upon an Extremely Rapid Increase in Acquisition Probability that Occurs Just Prior to Collision

In cases where visual acquisition is very difficult, the predicted probability of visual acquisition remains very low until a few seconds before collision and then rapidly climbs to unity. This is because the SEE calculations assume that search effort is constant and that the range will go to zero at collision. Under these assumptions, the visual size of the target becomes so large that visual acquisition is certain to occur before zero range is reached, even if only in the last tenth of a second before collision. In reality, two things prevent this. The first is that the range to the pilot's eyes may not be zero, even at collision. But more importantly, visual search effort is not really continuous (as the model assumes), but occurs intermittently as the pilot's visual scan moves from inside to outside the cockpit. The actual probability of acquisition will lag the model prediction during periods when the predicted probability is rapidly increasing. Thus, the model prediction that the probability of acquisition will shoot up to unity in the last few seconds before collision is an artificial result for most actual flight conditions. This is not normally a problem, since most analyses will discount visual acquisitions that occur so late that effective avoidance is unlikely to be achieved.

3. GENERATING A SEE1 ANALYSIS

This section provides the information needed to run the basic visual aquisition program, SEE1. A description of the computational techniques employed in SEE1 is provided in Section 4.

SEE1 computes tables containing the probabilities of visual acquisition, PACQ, for a subject aircraft that encounters a target aircraft under specified visual search conditions. The search conditions are specified by a set of 16 input variables. If tables for several sets of conditions are of interest, they must be computed as separate cases.

For some purposes, multiple runs of SEE1 are necessary. For example, if the probability of either aircraft acquiring is desired, then SEE1 must be run twice, with one aircraft being the subject in the first run and the target in the second run. SEE1 does not provide averaging over several tables, nor does it provide random selection of input variables.

3.1 Row and Column Variables

The PACQ tables are created by varying two user-selected input variables. The variable that is assigned anew for each row is called the row variable. The variable that is assigned anew for each column is called the column variable. The other 14 variables are constant for all entries in the table. For example, SEE1 could produce a table of PACQ values for various values of subject airspeed and visual range. For all values in this table, other variables (such as the airspeed of the target aircraft) would be constant.

The methods used to assign values to the row and column variable differ. The row variable is stepped from a minimum value to a maximum value with a fixed step interval. The column variable is stepped through fixed values input by the user. Up to 8 values are permitted. They need not be spaced evenly or be monotonic.

Figure 3.1 depicts the basic input/output structure of SEE1. The sixteen (16) input variables that must be specified in order to compute a single value of the acquisition probability (PACQ) are listed as variables 1 through 16 in Table 3.1. Each variable has a nominal value that is employed unless the user specifies otherwise. In addition to the 16 input variables, there are certain internal variables that are important at intermediate stages of the calculation. These variables are also listed in Fig. 3.1 and Table 3.1. The details of the computational modules are discussed in Section 4. Figure 3.2 illustrates how the 16 input variables can be determined from a description of a particular encounter scenario.

3.2 Input Variables

The following paragraphs provide a brief description of each of the 16 input variables. A guide to selecting proper values of variables is provided in Section 6.

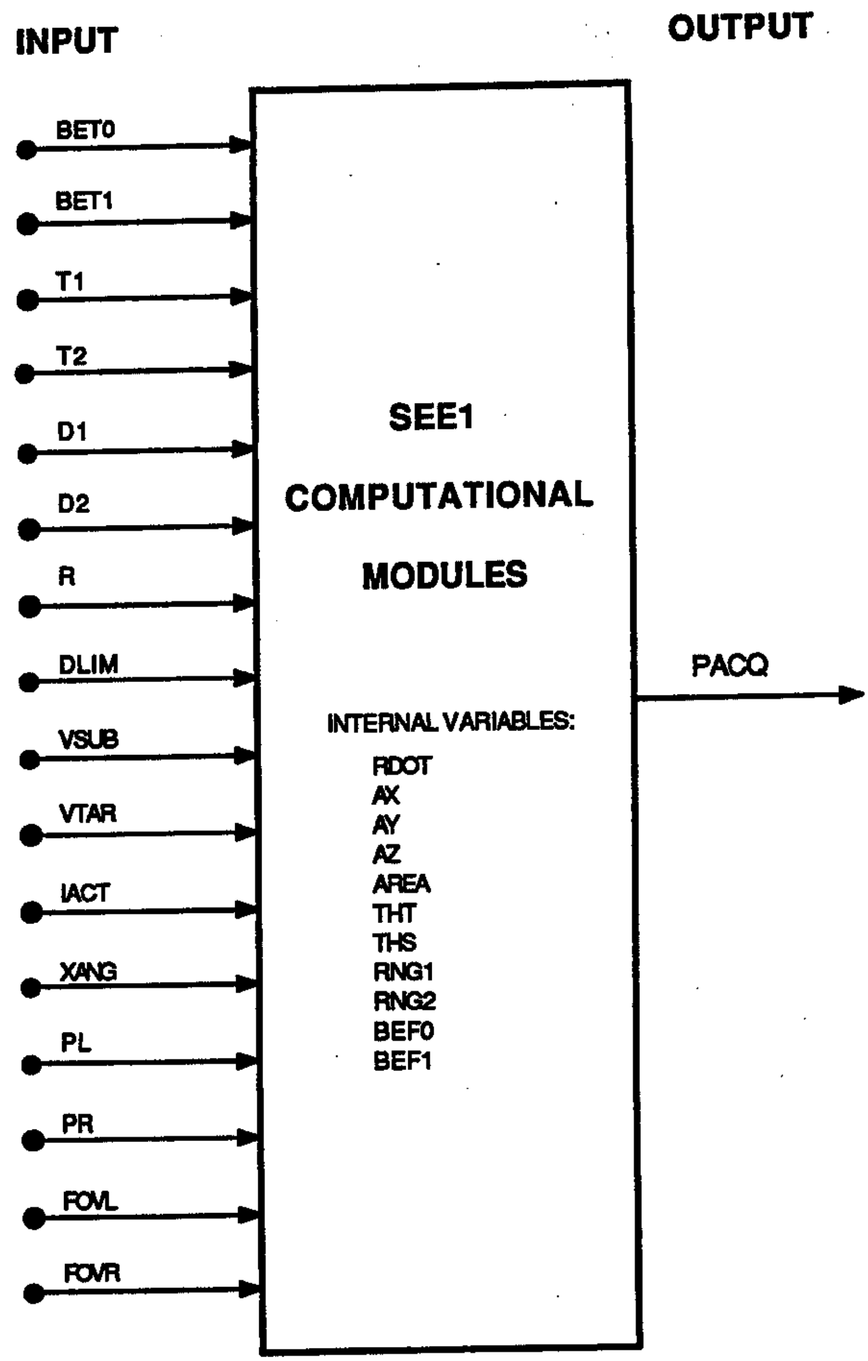

Figure 3.1 Basic input/output structure of SEE1. Sixteen input variables must be defined to compute the cumulative probability of visual acquisition (PACQ). See Table 3.1 for variable definitions. See Fig. 4.1 for a more detailed diagran of the modular structure.

TABLE 3.1

VARIABLES USED IN PROGRAM

NO.	SYMBOL	NAME	NOMINAL VALUE	DEFINITION
1	-	BETO	17000 per steradian-sec	Single-pilot β value before transition.
2	-	BET1	17000 per steradian-sec	Single-pilot β value after transition.
3	t_1	T1	180 sec	Modified tau time for β transition (time to closest approach).
4	t_2	T2	12 sec	Modified tau time for evaluation of PACQ (time to closest approach).
5	D1	D1	0. nmi	Modified tau range parameter for β transition.
6	D2	D2	0. nmi	Modified tau range parameter for PACQ evaluation.
7	R	R	20. nmi	Visual range.
8	d	DLIM	1 arc-min	Resolution limit of human eye.
9	V_1	VSUB1	180 kt	Airspeed of subject aircraft.
10	V_2	VTAR	130 kt	Airspeed of target aircraft.
11	-	IACT	1	Aircraft type for target aircraft (see table).
12	χ	XANG	180. deg	Crossing angle (relative heading).
13	PL	PL	1.0	Search intensity for pilot in left seat. (0 = no pilot 1 = normal search)
14	PR	PR	0.0	Search intensity for pilot in right seat. (0 = no pilot search, 1 = normal search)
15	FL	FOVL	-120 deg	Field-of-view cutoff on left side for searching pilot in left seat (measured positive clockwise from straight ahead).

TABLE 3.1

VARIABLES USED IN PROGRAM (CONT'D)

NO.	SYMBOL	NAME	NOMINAL VALUE	DEFINITION
16	FR	FOVR	90 deg	Field-of-view cutoff on right side for searching pilot in left seat (measured positive clockwise from straight ahead).
17	$\dot{r}$	RDOT	380 kt*	Range rate of target aircraft.
18	A_x	AX	35 sq.ft.*	Head-on visual area of target.
19	A_y	AY	85 sq.ft.*	Broadside visual area of target.
20	A_z	AZ	260 sq.ft.*	Topside visual area of target.
21	A	AREA	35 sq.ft.*	Final visual area of target.
22	θ_T	TH1	0 deg*	Bearing of target aircraft as seen from subject aircraft.
23	θ_S	TH2	0 deg*	Bearing of subject aircraft as seen from target aircraft.
24	R_1	RNG1	3.78 nmi*	Range at which β transition occurs.
25	R_2	RNG2	1.27 nmi*	Range at which PACQ is evaluated.
26	β_0	BEF0	17000* per steradian-sec	Effective value of β before transition.
27	β_1	BEF1	17000* per steradian-sec	Effective value of β after transition.
28	P	PACQ	-	Probability of visual acquisition.

*Value that is computed when all 16 input variables have nominal value.

ENCOUNTER DESCRIPTION	CONCLUSIONS
• The subject aircraft is flying at a speed of 180 knots.	• V1 = 180 kt
• The prevailing visual range is 20 nmi.	• R = 20 nmi
• There is one pilot in the left seat and one in the right seat. Normal visual acuity applies.	• PL = 1.0, PR = 1.0, DLIM = 1 arc-min
• The cockpit has a symmetric field-of-view. The pilot in the left seat can see all approach bearings from 8 o'clock around to 3 o'clock (i.e., from 120 degrees to the left of the nose of the aircraft to 90 degrees to the right).	• FOVL = 120 deg, FOVR = 90 deg
• The pilots are unaware of the presence of the target aircraft. Workload is low and pilots are expending a normal amount of effort on visual search.	• BET0 = 17000/ster-sec (the value for unalerted search)
• A target aircraft begins to approach. It is a Piper PA-28 flying on an unaccelerated collision course.	• IACT = 1
• It is flying at a speed of 130 knots.	• V2 = 130 kt
• It is on a head-on course (i.e., its heading differs by 180 degrees from that of the subject aircraft).	• XANG = 180 deg
• The crew of the subject aircraft receives a traffic advisory from their TCAS unit when the traffic is projected to be 40 seconds from reaching a range of 0.3 nmi.	• T1 = 40 sec, D1 = 0.3 sec
• Both crew members then concentrate on finding the indicated traffic.	• BET1 = 130000/ster-sec (the value for alerted search)
• Visual search is considered to have failed unless acquisition occurs prior to 12 seconds from collision	• T2 = 12 sec, D2 = 0.0 nmi

Figure 3.2 Parameter Selection Example

(1) Input Variable BETO.

SEE1 allows the single-pilot value of model parameter β to transition once from an early value to a final value. This is used to model the effect of receiving a traffic advisory. BETO is the early value of β, prior to receipt of any traffic advisory. See further description of the transition logic in Section 4.2.6.

(2) Input Variable BET1

This is the final value of β that applies after the transition event. If it is set equal to BETO, then the timing of any transition does not affect visual search. See further descripton in Section 4.2.6.

(3) Input Variable T1

The β transition occurs T1 seconds prior to the time the aircraft will reach a range of D1. See further discussion in Section 4.2.6.

(4) Input Variable T2

The probability of visual acquisition (PACQ) output by SEE1 is the cumulative probability of acquisition at T2 seconds prior to the target aircraft reaching a range of D2 nautical miles. See further discussion in Section 4.2.5.

(5) Input Variable D1

D1 is the range threshold criteria associated with the β transition. It is used in conjunction with T1 to determine the time of β transition. See further discussion in Section 4.2.6.

(6) Input Variable D2

D2 is the range threshold criteria associated with the point at which the probability of visual acquisition is to be evaluated. See further discussion in Section 4.2.5.

(7) Input Variable R

R is the prevailing visual range (nmi) along the line of sight to the target aircraft. This is the range at which any target, regardless of its size and contrast, tends to become invisible to the human eye. It is defined technically as the range at which the atmospheric conditions reduce the contrast of a target to 5 per cent of its inherent (close range) contrast. See Appendix A for a description of how R enters into the calculation of the acquisition rate.

(8) Input Variable DLIM

DLIM is the assumed resolution limit (in arc-min) of the human eye. SEE1 assigns a zero acquisition rate to targets beyond this limit. The nominal value of DLIM is 1.0 arc-min. Normally, this variable has little impact upon the calculated PACQ values since most of the opportunity to acquire a closing target comes after the subtended angle of the target has become substantially larger than the resolution limit. See further discussion in Section 4.2.7.

(9) Input Variable VSUB

VSUB is the airspeed of the subject aircraft. Note that SEE1 does not include a wind model, hence all variables are defined in the airmass coordinate system.

(10) Input Variable VTAR

VTAR is the airspeed of the target aircraft.

(11) Input Variable IACT

IACT is the index to target aircraft type as defined in the table of aircraft types from file SEEAC (see Appendix C). For example, IACT=1 selects a PA-28 target; IACT=2 selects a Boeing 727 target.

Note that the type of the subject aircraft need not be defined, since the physical size of the subject aircraft is irrelevant to its ability to acquire the target. However, the speed of the subject aircraft (VSUB) is needed to calculate the rate at which the target is closing upon the subject.

A new aircraft type can be inserted at run time by simply typing "-1" in place of a defined IACT value. SEE1 will then ask for the name of the new aircraft type and for its three principal areas, AX, AY, and AZ (see Section 4.2.3 for discussion of how the principal areas are used in the calculations). A new aircraft type defined in this way is temporary, and will not be retained for use in subsequent sessions. In order to permanently enter a new aircraft type in the aircraft list, the aircraft description must be entered in the disk file of aircraft types (SEEAC). Any text editor that does not insert word-processing control codes into a text file can be used for this purpose (see Appendix C).

For a new aircraft type, values of the principal areas can be found by tracing the aircraft silhouettes onto a sheet of gridded graph paper, counting the number of grid squares within each silhouette, and multiplying the total by the number of square feet in a single grid square.

(12) Input Variable XANG

XANG is the crossing angle (relative heading) of the target relative to the subject aircraft in the airmass coordinate system. XANG=180° corresponds to a head-on encounter. XANG=0° corresponds to one aircraft overtaking the other from the six o'clock position. See Section 4.2.2 for further discussion of XANG.

(13) Input Variable PL

PL is equal to 1 if there is a pilot in the left seat of the subject aircraft. PL is equal to 0 if there is no pilot in this seat.

(14) Input Variable PR

PR is equal to 1 if there is a pilot in the right seat of the subject aircraft. PR is equal to 0 if there is no pilot in this seat.

(15) Input Variable FOVL

FOVL is the bearing of the left-side visibility cut-off for the field-of-view from the left seat of the subject aircraft. It is measured in degrees clockwise from the nose of the aircraft (thus, FOVL=-120° corresponds to the 8 o'clock bearing position). See Section 4.2.4 for further description.

(16) Input Variable FOVR

FOVR is the bearing of the right-side visibility cut-off for the field-of-view from the left seat of the subject aircraft. It is measured in degrees clockwise from the nose of the aircraft (thus, FOVR=90° corresponds to the 3 o'clock bearing position).

It should be noted that the field-of-view parameters (FOVL and FOVR) do not change automatically as the aircraft type is changed. They must be set independently if different values are desired because of different aircraft types.

3.3 Steps in Generating a SEE1 Analysis

Figure 3.3 provides a set-up form that can be used in assembling the information required for a SEE1 analysis. A guide to selection of values for input variables can be found in Section 6. Figure 3.4 provides a summary of the steps involved in generating a SEE1 analysis.

SEE1 SET-UP FORM

RUN TITLE ____________________ DATE __________

NOTE ____

TARGET AIRCRAFT : IACT _____ (NAME _______ AX ____ AY _____ AZ _____)

ROW VARIABLE _______ MIN _____ MAX ______ STEPS _____

COLUMN VARIABLE _________ VALUES _____ _____ _____ _____ _____

_____ _____ _____

INPUT VARIABLES:

BET0 ______________
(β prior to alert) (17000/ster-sec)

BET1 ______________
(β after alert) (17000/ster-sec)

T1 ______________
(tau for alert) (40 sec)

T2 ______________
(tau for PACQ evaluation) (12 sec)

D1 ______________
(range parameter for alert) (0.0 nmi)

D2 ______________
(range parameter for PACQ) (0.0 nmi)

R ______________
(visual range) (20 nmi)

DLIM ______________
(resolution limit of eye) (1.0 arc-min)

VSUB ______________
(airspeed of subject AC) (180 kt)

VTAR ______________
(airspeed of target AC) (130 kt)

IACT ______________
(target aircraft type) (1=PA28)

XANG ______________
(crossing angle) (180 deg)

PL ______________
(left seat pilot) (1.0)

PR ______________
(right seat pilot) (0.0)

FOVL ______________
(left side FOV cut-off) (-120 deg)

FOVR ______________
(right side FOV cut-off) (90 deg)

Figure 3.3 Set-up form for SEE1 analysis.

(hb.fig3.4)

SEE1 ANALYSIS : STEP-BY-STEP

STEP 1 : Select values for the input variables (complete a SEE1 set-up form).

STEP 2 : Insert diskette and type "SEE1"

STEP 3 : Select the row variable (by typing its name).

STEP 4 : Specify the minimum, maximum, and step size for the row variable.

STEP 5 : Select the column variable (by typing its name).

STEP 6 : Type in the values for the column variable (maximum of 8 different values)..

STEP 7 : Select the target aircraft type from the displayed list.

Note : If no type in the list is satisfactory, then type "-1". The program will then allow you to define a new type.

STEP 8 : Check the input variables displayed on the screen. If any is incorrect, type its name. At the prompt, type the new value.

STEP 9 : If all input variables are correct, type CR. At the prompt, enter a note. The computations will then begin.

STEP 10 : After the analysis, direct results to either the printer or the console display.

Note : Results are also stored in text form in a disk file SEE1.OUT. If you wish to save this file, you must exit SEE1 and rename the file prior to your next analysis.

Figure 3.4 Step-by-step instructions for generating a SEE1 analysis.

4. SEE1 ANALYSIS PROGRAM

This section provides a description of the computational structure of SEE1. Although it is possible to run SEE1 without understanding this structure, the interpretation of SEE1 outputs is aided by familiarity with the following material.

4.1 Basic Computational Structure of SEE1

SEE1 employs a number of modules (subroutines) that, when called in the proper order, compute a probability of visual acquisition (PACQ) for a given set of search conditions. The search conditions are defined in terms of the set of 16 input variables shown earlier in Table 3.1. The program must be given all 16 input variables in order to compute a single value of PACQ. Each variable has a nominal value that is employed unless the user inserts a non-nominal value.

The block in Fig. 3.1 entitled "SEE1 Computational Modules" consists of seven component modules. A more detailed flowchart that shows these modules is provided in Fig. 4.1. The 16 input variables are all shown in this figure as external inputs to the modules in which they are used. In addition, 11 computed variables are shown as outputs of the modules in which they are computed. These computed variables are listed in Table 3.1 as variables number 17-27.

One input variable is selected to be the row variable. It is stepped from a minimum value (xmin) to a maximum value (xmax) with a constant step interval (xstep). The column variable is not incremented in this way -- instead the user must type in the actual values to be used. A maximum of 8 values is permitted. This limit is due to the limited amount of output that can be placed in a single plot or in a single printed table.

Figure 4.2 is a flowchart that depicts the manner in which the loops that control the output are nested.

4.2 Description of SEE1 Modules

The computations carried out in each of the seven basic SEE1 program modules is described below. Each module corresponds to a single Pascal procedure. A listing of the SEE1 code for the modules can be found in Appendix C.

4.2.1 Module 1: Aircraft Type Look-up

Module 1 uses the aircraft type number, IACT, to determine various characteristics of the target aircraft. Table 4.1 lists the aircraft types that are provided in the disk file SEEAC.A00. Each aircraft type is associated with an abbreviated name (e.g., PA28) and three numbers (AX, AY, and AZ) that define the visual area of the aircraft as projected upon the three principal coordinate planes (see Fig. 4.3). A new aircraft type can be created by assigning table values from the console.

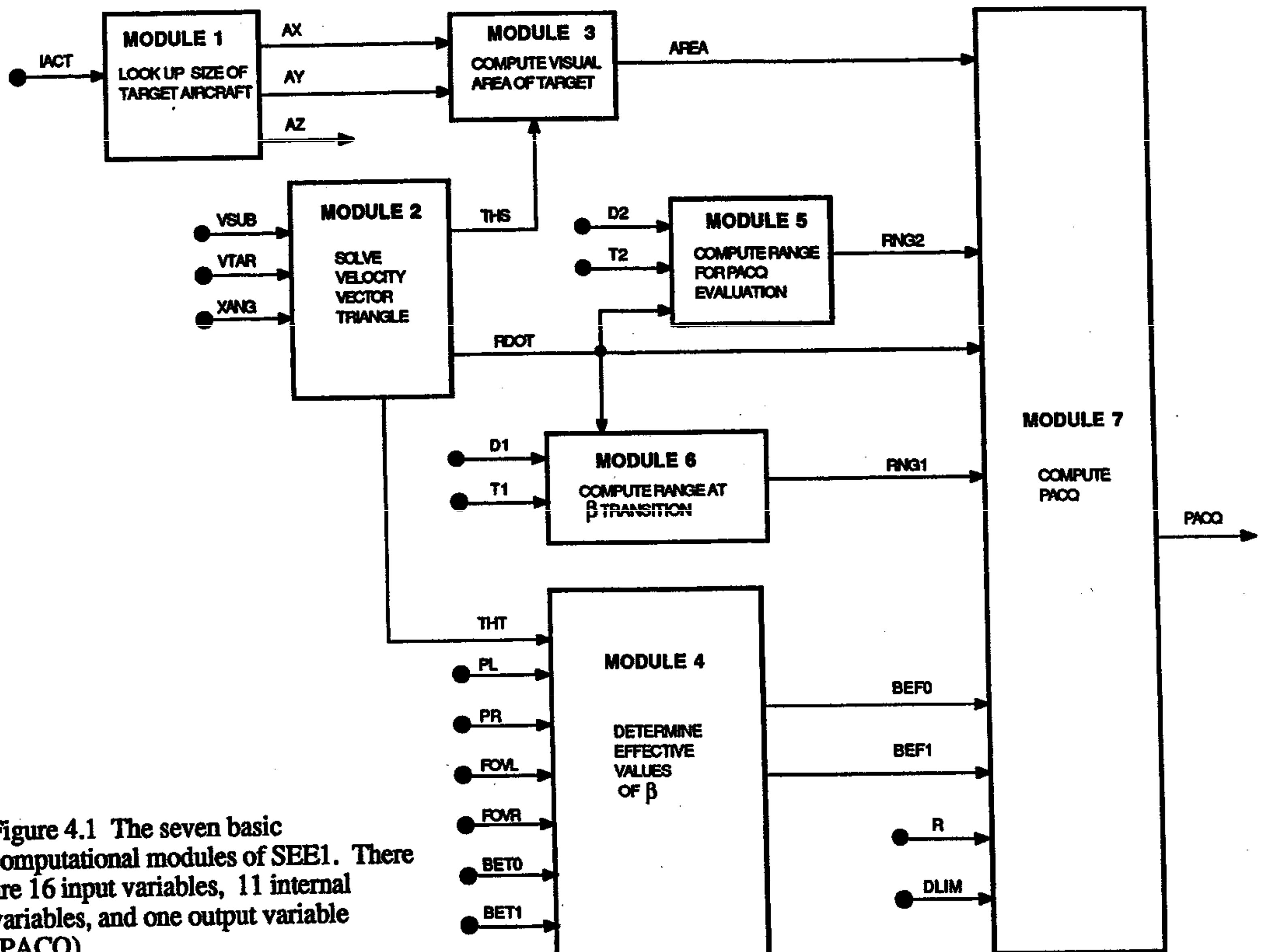

Figure 4.1 The seven basic computational modules of SEE1. There are 16 input variables, 11 internal variables, and one output variable (PACQ).

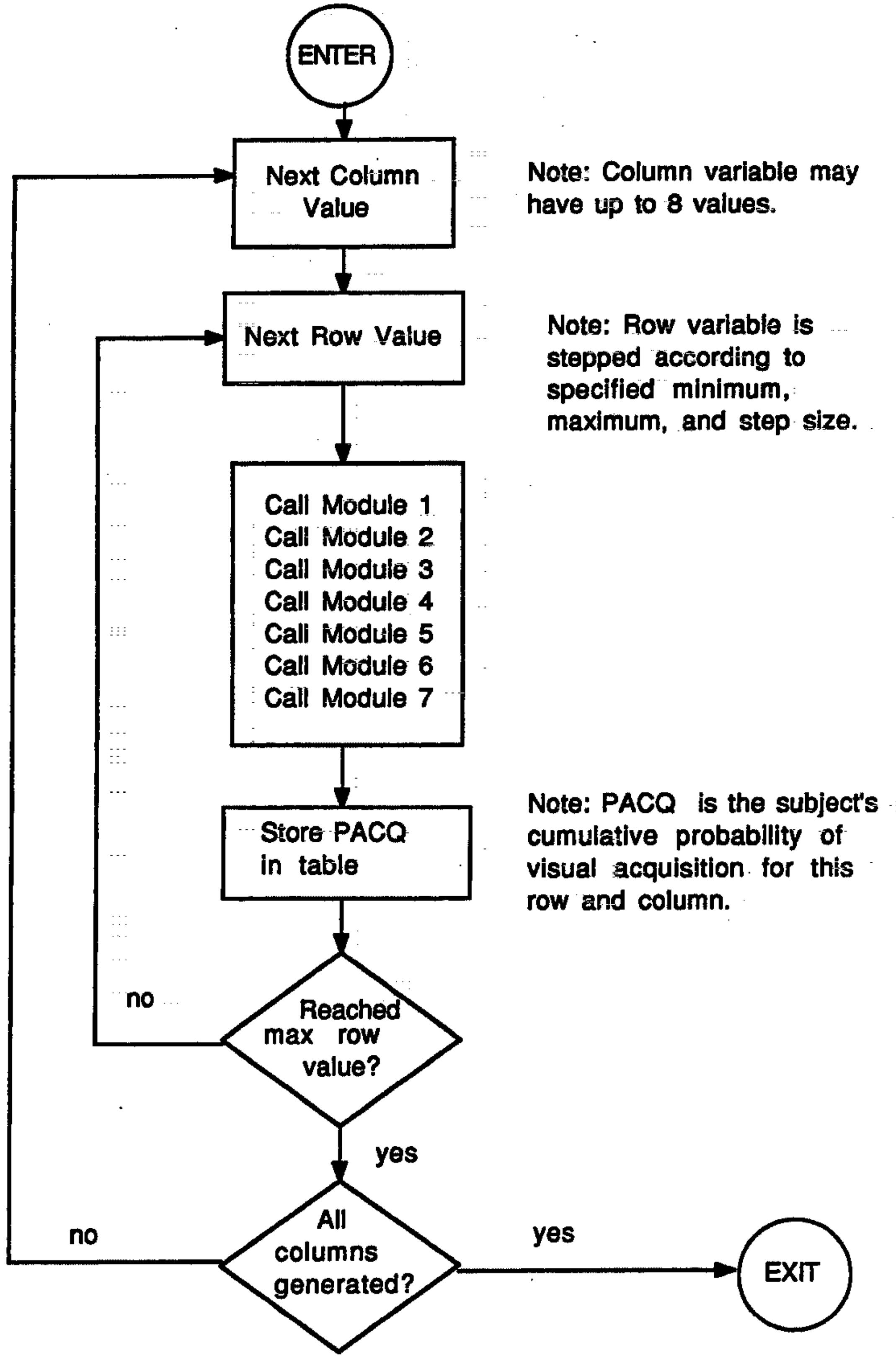

Figure 4.2 Loops controlling table output for SEE1.

TABLE 4.1

AIRCRAFT TYPES USED IN SEE1

IACT	AIRCRAFT TYPE	NAME	AX (sq. ft.)	AY (sq. ft.)	AZ (sq. ft.)
1	Piper PA-28	PA28	35	85	260
2	Boeing 727	B727	400	1900	3100
3	Boeing 747	B747	1200	5700	9300
4	McDonnell-Douglas DC9	DC9	300	1425	2325
5	Cessna 421	C421	81	171	417
6	U-21 (King Air)	U21	127	267	600
7	F-18	F18	50	280	540

(Note: New aircraft types can be created by editing the SEEAC file. See Appendix B for further information.)

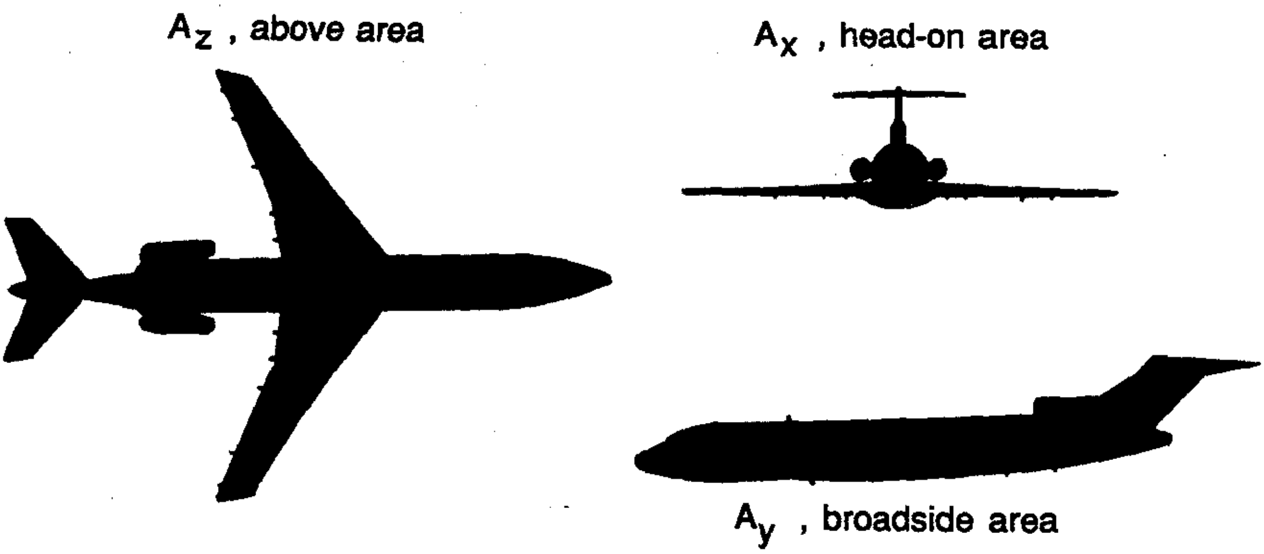

Figure 4.3 Principal target areas used to compute the visual area of the target aircraft as seen from the subject aircraft.

4.2.2 Module 2: Solving the Velocity Vector Triangle

From a knowledge of the aircraft airspeeds and the crossing angle, Module 2 solves the velocity vector triangle (Fig. 4.4) to determine the approach bearings and range rate. This solution assumes that the aircraft are on a rectilinear collision course and hence that the horizontal miss distance is zero. For these conditions, the values of the range rate (RDOT) and the approach bearings (THS and THT) are constant.

Approach bearings are measured positive clockwise from straight ahead and are expressed as a number between -180 degrees and +180 degrees. The crossing angle, XANG, is simply the relative heading formed by subtracting the heading of the target aircraft from the heading of the subject aircraft. The simple trigonometric formula that relates the approach bearings and the crossing angle is

$$XANG = 180° + THS - THT$$

The approach bearing for the target aircraft (THT) is important since it determines the visual area that will be seen from the subject aircraft (see discussion in 4.2.3). The approach bearing for the subject (THS) is used in Module 4 to determine whether or not the target is within the cockpit field-of-view.

4.2.3 Module 3: Computing the Visual Area of the Target

The visual area of the target is computed by projecting the principal visual areas AX and AY onto a plane that is normal to the line-of-sight between the two aircraft. This plane is at an angle THT with respect to the Y-Z plane for the target aircraft. An approximate correction for shielding is applied by assuming that the actual visual area is the largest of the projections of AX and AY plus one-third of the smallest projection (see Reference 2). This approximation is errorless when the aircraft is viewed along one of the principal axes. Because Module 3 assumes that the line-of-sight lies within the X-Y plane, the topside area, AZ, is not allowed to contribute to the target visual area. This is normally a good assumption, but can result in underestimation of the visual area if the pitch angle of the aircraft is large. AZ is defined in the module to allow for future extension of the software.

4.2.4 Module 4: Determining the Effective Values of β

Module 4 examines cockpit visibility and crew complement in order to determine their effect upon search effectiveness. It does so by taking the nominal single-pilot β values and summing them to produce effective β values. For a single pilot devoting normal effort to visual search, the value of β before and after the transition event (see 4.2.6) are represented by parameters BETO and BET1. Module 4 computes two corresponding effective β values, BEFO and BEF1, by summing the β values for each pilot who is able to contribute to visual search.

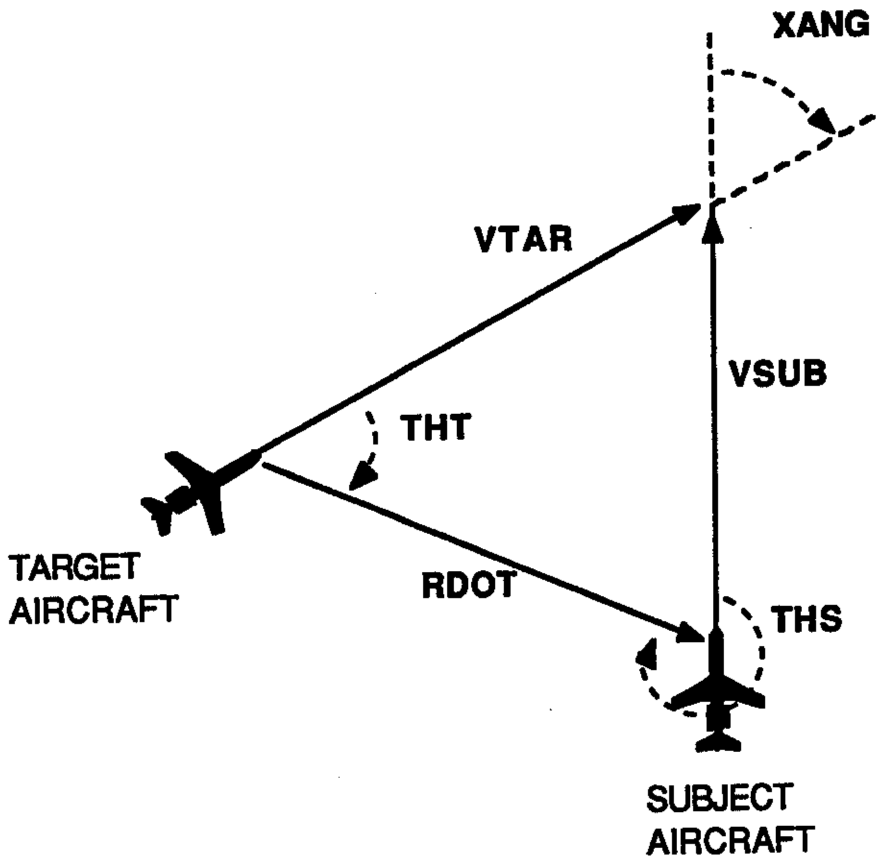

Figure 4.4 The velocity vector triangle is solved in Module 2 in order to determine the closing rate and the bearings of approach.

The field-of-view of the cockpit is determined by the variables FOVL and FOVR (see Fig. 4.5). It is assumed that the cockpit visibility is symmetric and that there is no cut-off in elevation (normally true for targets on collision courses). When the target is outside the field-of-view, the effective β value is set to zero.

By setting PL and PR, the user specifies the number of pilots that contribute to search for both left and right visibility limits. Normally PL and PR are either zero (no pilot in the seat) or unity (single pilot in seat and searching normally). It is possible to use a value between 0 and 1 to represent a pilot who is present, but is devoting less than the normal amount of time to visual search. Or a value greater than 1 can be used to represent search by an additional crew member.

For the nominal input values given in Table 3.1, it is assumed that a single pilot is searching from the left seat of the aircraft (PL=1.0, PR=0.0). For the nominal values of crossing angle (XANG=180 degrees), the intruder is always within the field of view. Thus for the nominal input values, BEF0=BET0 and BEF1=BET1.

4.2.5 Module 5: Determining the Range of Evaluation

The cumulative probability of visual acquisition, PACQ, depends upon the range, RNG2, at which PACQ is to be evaluated. Often it is desirable to set this range to the latest point at which sufficient time exists for evasive action. Then PACQ will represent the probability of visual acquisition with sufficient lead time for avoidance.

RNG2 is computed using a modified tau criterion based upon two input parameters D2 and R2:

$$RNG2 = D2 - RDOT*T2$$

From this equation it can be seen that RNG2 is reached when the target is T2 seconds from reaching a range of D2. With this formulation, RNG2 can be determined as either a constant range, a constant time-to-collision, or a modified tau criterion. For example, if T2=0, then the equation yields a simple range criteria for which PACQ is evaluated at range D2. If D2=0, then a simple time criteria results in which PACQ is evaluated when T2 seconds remain before collision. If neither T2 nor D2 are zero, then a modified tau criterion results.

4.2.6 Module 6: Determining the Range of β Transition

In a particular scenario, the nominal value of β for a single pilot may be altered by an event such as the receipt of a traffic advisory. SEE1 allows the nominal single-pilot value of β to transition from one value to another at a predefined point during the encounter. The initial value of β is BET0. After the transition, the value of β is BET1 (see Fig. 4.6). The range at which transition occurs is RNG1. It is computed using a modified tau criterion involving the two input parameters D1 and R1:

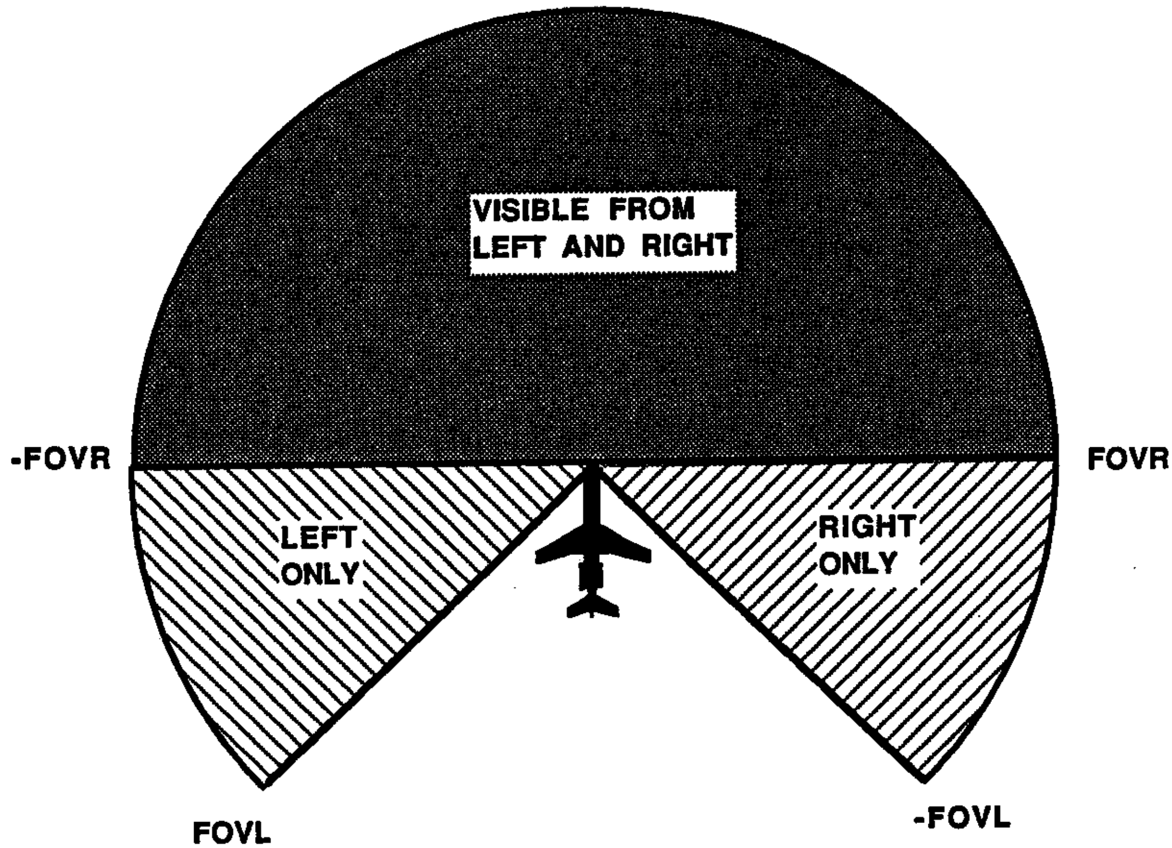

Figure 4.5 Cockpit field-of-view model for visual search from left and right seats in cockpit. FOVL and FOVR are the bearing angles (in degrees clockwise from 12 o'clock) at which the field-of-view terminates for a pilot in the left seat.

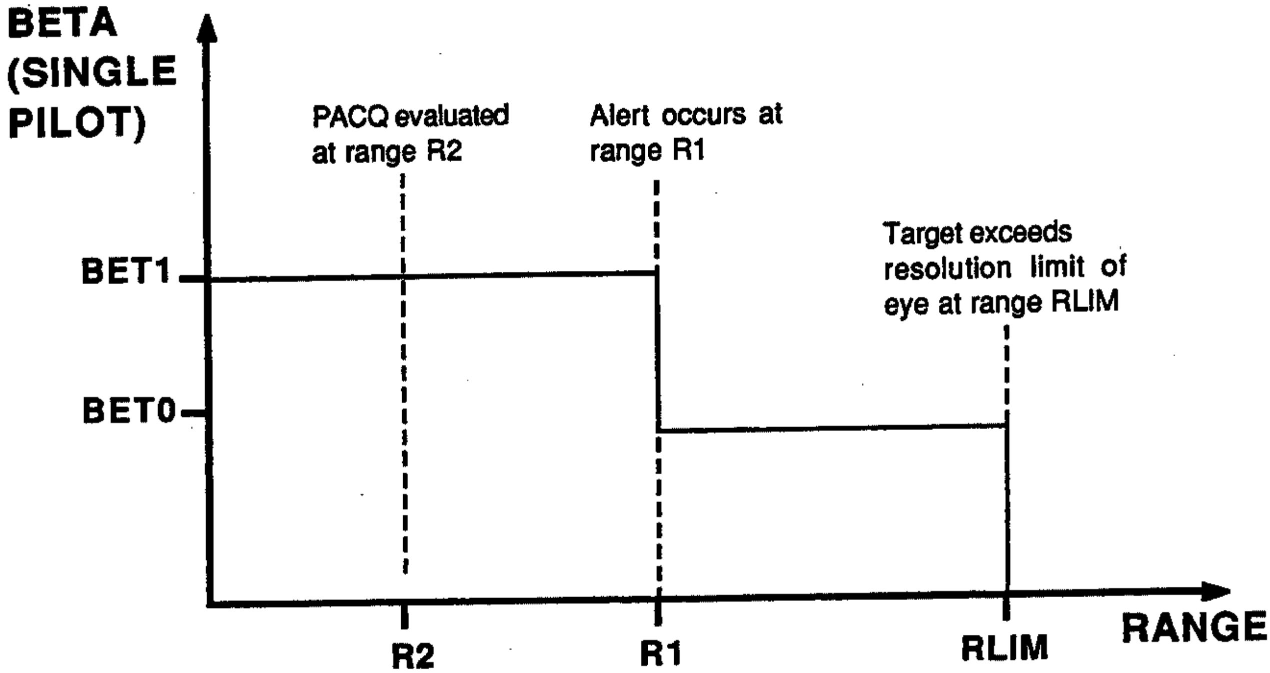

Figure 4.6 In SEE1, the effective value of beta for a single pilot is either zero, BET0, or BET1. The transition from zero to BET0 occurs at range RLIM (the resolution limit of the eye). The transition to BET1 occurs at the range at which a traffic alert is received.

RNG1 = D1 - RDOT*T1

The transition occurs when the target is T1 seconds from reaching a range D1. Note that if T1=0, then a simple range criterion results. If D1=0, then a simple time-to-collision criterion results.

If the application of interest does not require a transition, the transition is easily avoided by setting BET1 equal to BET0. The settings of D1 and T1 then have no effect upon the computations (because the values of β are unchanged by reaching the transition point).

4.2.7 Module 7: Computing the Cumulative Probability of Acquisition PACQ

Module 7 is the final module that actually computes the probability of visual acquisition, PACQ. It requires 8 inputs as shown in Fig. 4.1.

Module 7 assumes that the acquisition rate drops immediately to zero when the target is beyond a range, RLIM, corresponding to the resolution limit of the human eye. RLIM is defined as follows: First, DLIM, AREA, and RDOT are used to compute the range at which the solid angle subtended by the target equals that of a circle with diameter DLIM. This range is given by the formula

1.1284*SQRT(AREA)/DLIM

where DLIM is in radians. Then, if this range is greater than R, the visual range, it is reduced to R. The net result is

RLIM = minimum [1.1284*SQRT(AREA)/DLIM, R]

A more sophisticated model for RLIM could be constructed, but in most analyses, only a tiny fraction of the opportunity for visual acquisition occurs before the target is well within visual range and well above the resolution threshold. Hence, the details of the model in this regard are normally of little consequence.

Note that if RLIM is less than RNG1, then β will already have transitioned to BET1 before the target is close enough to be seen. In this case, the value of BET0 has no affect upon the calculations.

4.3 SEE1 Analysis Results

Examples of SEE1 output for some typical cases are provided in this section.

Figure 4.7 contains SEE1 output in which T2, the time of evaluation, is the row variable. Because D2=0, T2 is the time to projected collision. The analysis shows how the cumulative probability of visual acquisition

SEE1 VISUAL ACQUISITION ANALYSIS

NOTE :Example.
SEE1.PAS (Version A01d)
InFNam= G:SEE1SET.A00 VFNam= G:SEEVAR.A00
ACFNam= G:SEEAC.A00 OutFNam= G:SEE1.OUT

INITIAL VALUES OF INPUT VARIABLES:

BET0	17000.0	R	2.00*	PL	1.00
BET1	17000.0	DLIM	1.00	PR	0.00
T1	180.0	VSUB	180.0	FOVL	-120.00
T2	6.00*	VTAR	130.0	FOVR	90.0
D1	0.00	IACT	1.00		
D2	0.00	XANG	180.0		

* = value on first iteration

ROW VARIABLE:
T2 is varied from 6.00 to 60.0 in steps of 6.00
COLUMN VARIABLE :
R assumes values 2.00 3.00 5.00 15.00
30.00 9999.00

Target aircraft is PA28
AX= 35.0 sq ft AY= 85.0 sq ft AZ= 260.0 sq ft AREA= 35.0 sq ft
RDOT= -310.0 kt THT= -0.0 deg THS= 0.0 deg
PACQ TABLE :

TABLE OF PACQ VALUES

	R					
T2	2.00	3.00	5.00	15.00	30.00	9999.00
6.0	0.0720	0.1082	0.1525	0.2204	0.2430	0.2684
12.0	0.0113	0.0247	0.0461	0.0878	0.1039	0.1232
18.0	0.0020	0.0076	0.0184	0.0439	0.0549	0.0687
24.0	*	0.0025	0.0082	0.0235	0.0307	0.0401
30.0	*	0.0006	0.0036	0.0123	0.0167	0.0226
36.0	*	*	0.0014	0.0055	0.0077	0.0108
42.0	*	*	0.0002	0.0011	0.0015	0.0022
48.0	*	*	*	*	*	*
54.0	*	*	*	*	*	*
60.0	*	*	*	*	*	*
avg.	0.0085	0.0144	0.0230	0.0394	0.0458	0.0536

--*-* END OF ANALYSIS *-*-*-*

Figure 4.7. SEE1 Analysis Results. A PA28 target aircraft flying at 130 knots approaches in a head-on geometry. The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at times from 6 to 60 seconds prior to collision.

grows as the aircraft approach each other. Visual range, R, is the column variable. Six values are chosen. The value of 9999 nmi was used to approximate a perfectly clear atmosphere. All variables other than T2 and R have the nominal values.

For cases in which the target aircraft is outside the field-of-view of all crew members, SEE1 will write "xfov" in the table instead of "0.0000". This serves to explain the reason for zero probabilities that may appear unexpectedly.

It can be seen from the table that if the visual range were only 3 nmi, then the predicted probability of visual acquisition at 12 seconds to collision would be only 2.47 percent. For a perfectly clear day, the probability would increase to only 12.32 percent. Thus, visual acquisition is very difficult for the defined encounter conditions.

Figure 4.8 is a similar analysis, but the target aircraft is now a Boeing 727 flying at 240 knots. The closing rate increases by 60 knots, but because the target is so much larger, it is easier to acquire. The probability of visual acquisition for a perfectly clear day is now 63 percent at 12 seconds to collision.

Figure 4.9 is a SEE1 analysis in which the row variable is the crossing angle (XANG). The PACQ table shows that the crossing angle can have an important impact upon the probability of visual acquisition.

Additional examples of SEE1 analyses are provided in Figs. 4.10 through 4.12.

```
SEE1 VISUAL ACQUISITION ANALYSIS
     NOTE :Sample SEE1 output.
     SEE1.PAS (Version A01d)
     InFNam=      G:SEE1SET.A00        VFNam= G:SEEVAR.A00
     ACFNam=     G:SEEAC.A00         OutFNam= G:SEE1.OUT

INITIAL VALUES OF INPUT VARIABLES:
    BET0      17000.0        R            2.00*         PL            1.00
    BET1      17000.0        DLIM         1.00          PR            0.00
    T1          180.0        VSUB        180.0          FOVL       -120.00
    T2            6.00*      VTAR        240.0          FOVR          90.0
    D1            0.00       IACT          2.00
    D2            0.00       XANG        180.0

              * = value on first iteration

ROW VARIABLE:
     T2   is varied from      6.00 to      60.0 in steps of     6.00
COLUMN VARIABLE :
     R    assumes values      2.00    3.00    5.00   15.00
                             30.00 9999.00
  Target aircraft is B727
  AX= 400.0 sq ft    AY=1900.0 sq ft    AZ=3100.0 sq ft    AREA=  400.0 sq ft
  RDOT= -420.0 kt     THT=   -0.0 deg    THS=     0.0 deg
PACQ TABLE :
```

TABLE OF PACQ VALUES

T2	R = 2.00	3.00	5.00	15.00	30.00	9999.00
6.0	0.2617	0.4085	0.5703	0.7719	0.8265	0.8812
12.0	0.0285	0.0854	0.1894	0.4220	0.5170	0.6332
18.0	*	0.0191	0.0709	0.2445	0.3367	0.4660
24.0	*	0.0021	0.0286	0.1515	0.2312	0.3556
30.0	*	*	0.0115	0.0986	0.1650	0.2788
36.0	*	*	0.0039	0.0665	0.1210	0.2226
42.0	*	*	0.0003	0.0460	0.0905	0.1797
48.0	*	*	*	0.0324	0.0685	0.1461
54.0	*	*	*	0.0231	0.0522	0.1189
60.0	*	*	*	0.0167	0.0399	0.0966
avg.	0.0290	0.0515	0.0875	0.1873	0.2449	0.3379

--*-* END OF ANALYSIS *-*-*-*

Figure 4.8. SEE1 Analysis Results. A B727 target aircraft flying at 240 knots approaches in a head-on geometry. The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at times from 6 to 60 seconds prior to collision.

```
SEE1 VISUAL ACQUISITION ANALYSIS
    | NOTE :Sample SEE1 output.
    | SEE1.PAS (Version A01d)
    | InFNam=     G:SEE1SET.A00       VFNam= G:SEEVAR.A00 .
    | ACFNam=    G:SEEAC.A00          OutFNam= G:SEE1.OUT

INITIAL VALUES OF INPUT VARIABLES:
    BET0      17000.0         R           2.00*        PL          1.00
    BET1      17000.0         DLIM        1.00         PR          0.00
    T1          180.0         VSUB      180.0          FOVL     -120.00
    T2           12.0         VTAR      130.0          FOVR        90.0
    D1           0.00         IACT        1.00
    D2           0.00         XANG        0.00*

              * = value on first iteration

ROW VARIABLE:
    XANG is varied from      0.00 to     180.0 in steps of    30.00
COLUMN VARIABLE :
    R    assumes values      2.00     3.00     5.00    15.00
                            30.00 9999.00
 | Target aircraft is PA28
 | AX=  35.0 sq ft    AY=  85.0 sq ft    AZ= 260.0 sq ft

PACQ TABLE :
```

TABLE OF PACQ VALUES

	R					
XANG	2.00	3.00	5.00	15.00	30.00	9999.00
0.0	0.9726	0.9863	0.9934	0.9975	0.9982	0.9987
30.0	0.8089	0.8895	0.9415	0.9780	0.9843	0.9895
60.0	0.2754	0.3934	0.5205	0.6816	0.7271	0.7739
90.0	0.0895	0.1536	0.2397	0.3808	0.4292	0.4841
120.0	0.0338	0.0659	0.1148	0.2051	0.2389	0.2792
150.0	0.0157	0.0332	0.0611	0.1153	0.1361	0.1612
180.0	0.0113	0.0247	0.0461	0.0878	0.1039	0.1232
avg.	0.3153	0.3638	0.4167	0.4923	0.5168	0.5443

```
*-*-*-* END OF ANALYSIS  *-*-*-*
```

Figure 4.9. SEE1 Analysis for Varying Crossing Angle (XANG). The subject aircraft is flying at 180 knots. A PA-28 target aircraft flying at 130 knots approaches at angles from 0 (subject overtakes) to 180 degrees (head-on). The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at 12 seconds prior to collision.

SEE1 VISUAL ACQUISITION ANALYSIS

SEE1.PAS Version A03
InFNam= G:SEE1IN.A00 VFNam= G:SEEVAR.A00
ACFNam= G:SEEAC.A00 OutFNam= G:SEE.OUT

INITIAL VALUES OF INPUT VARIABLES:

BETO	17000.0	R	2.00*	PL	1.00
BET1	17000.0	DLIM	1.00	PR	0.00
T1	180.0	VSUB	180.0	FOVL	-120.00
T2	12.0	VTAR	180.0	FOVR	90.0
D1	0.00	IACT	2.00		
D2	0.00	XANG	0.00*		

* = initial value (before iteration)

ROW VARIABLE: XANG is varied from 0.00 to 180.0 in steps of 30.00
COLUMN VARIABLE: R assumes values: 2.00 3.00 5.00 15.00 30.00 9999

Target aircraft is B727
AX= 400.0 sq ft AY=1900.0 sq ft AZ=3100.0 sq ft

TABLE OF PACQ VALUES

	R					
XANG	2.00	3.00	5.00	15.00	30.00	9999.00
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
30.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
60.00	0.9884	0.9993	1.0000	1.0000	1.0000	1.0000
90.00	0.6530	0.8683	0.9708	0.9988	0.9997	1.0000
120.00	0.2872	0.5255	0.7627	0.9539	0.9796	0.9943
150.00	0.1129	0.2523	0.4504	0.7459	0.8282	0.9060
180.00	0.0632	0.1516	0.2925	0.5546	0.6471	0.7508
avg.	0.5864	0.6853	0.7823	0.8933	0.9221	0.9501

--*-* END OF ANALYSIS *-*-*-*

Figure 4.10. SEE1 Analysis for Varying Crossing Angle (XANG). A B727 target aircraft flying at 180 knots approaches at angles from 0 (subject overtakes) to 180 degrees (head-on). The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at 12 seconds prior to collision.

```
SEE1 VISUAL ACQUISITION ANALYSIS
NOTE :Sample analysis.
     | SEE1.PAS Version  A03
     | InFNam= G:SEE1IN.A00        VFNam= G:SEEVAR.A00
     | ACFNam= G:SEEAC.A00         OutFNam= G:SEE.OUT

INITIAL VALUES OF INPUT VARIABLES:
    BET0     17000.0     R          2.00*    PL          1.00
    BET1     17000.0     DLIM       1.00     PR          0.00
    T1         180.0     VSUB      130.0     FOVL     -120.00
    T2          12.0     VTAR      130.0     FOVR        90.0
    D1          0.00     IACT       1.00
    D2          0.00     XANG       0.00*

             * = initial value (before iteration)

ROW VARIABLE: XANG is varied from     0.00 to    180.0 in steps of    30.00
COLUMN VARIABLE: R    assumes values :  2.00  3.00   5.00  15.00    30.00 9999
     | Target aircraft is PA28
     | AX=  35.0 sq ft   AY=  85.0 sq ft   AZ= 260.0 sq ft
```

TABLE OF PACQ VALUES

XANG \ R	2.00	3.00	5.00	15.00	30.00	9999.00
0.00	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
30.00	0.9831	0.9939	0.9980	0.9996	0.9998	0.9999
60.00	0.4459	0.5746	0.6942	0.8221	0.8538	0.8845
90.00	0.1525	0.2367	0.3392	0.4891	0.5367	0.5887
120.00	0.0600	0.1048	0.1670	0.2716	0.3085	0.3509
150.00	0.0314	0.0589	0.0990	0.1706	0.1967	0.2274
180.00	0.0234	0.0449	0.0767	0.1341	0.1552	0.1799
avg.	0.3852	0.4305	0.4820	0.5553	0.5787	0.6045

--*-* END OF ANALYSIS *-*-*-*

Figure 4.11. SEE1 Analysis for Varying Crossing Angle (XANG). The subject aircraft is flying at 130 knots. A PA-28 target aircraft flying at 130 knots approaches at crossing angles from 0 to 180 degrees (head-on). The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at 12 seconds prior to collision.

```
SEE1 VISUAL ACQUISITION ANALYSIS
     | SEE1.PAS Version  A03
     | InFNam= G:SEE1IN.A00        VFNam= G:SEEVAR.A00
     | ACFNam= G:SEEAC.A00         OutFNam= G:SEE.OUT

INITIAL VALUES OF INPUT VARIABLES:
    BETO       17000.0    R             2.00*    PL            1.00
    BET1       17000.0*   DLIM          1.00     PR            0.00
    T1           180.0    VSUB        180.0      FOVL       -120.00
    T2            12.0    VTAR        130.0      FOVR          90.0
    D1            0.00    IACT          1.00
    D2            0.00    XANG         90.0

            * = initial value (before iteration)

ROW VARIABLE: BET1 is varied from 17000.00 to 136000.0 in steps of 17000.00
COLUMN VARIABLE : R  assumes values :  2.00  3.00   5.00   15.00   30.00 9999
     | Target aircraft is PA28
     | AX=  35.0 sq ft   AY=  85.0 sq ft   AZ= 260.0 sq ft   AREA=   75.7 sq ft
     | RDOT= -222.0 kt    THT=  -35.8 deg    THS=   54.2 deg
```

TABLE OF PACQ VALUES

BET1	R 2.00	3.00	5.00	15.00	30.00	9999.00
17000.00	0.0895	0.1536	0.2397	0.3808	0.4292	0.4841
34000.00	0.1710	0.2835	0.4220	0.6166	0.6742	0.7339
51000.00	0.2452	0.3936	0.5606	0.7626	0.8140	0.8627
68000.00	0.3127	0.4867	0.6659	0.8530	0.8938	0.9292
85000.00	0.3742	0.5655	0.7460	0.9090	0.9394	0.9635
102000.00	0.4302	0.6322	0.8069	0.9436	0.9654	0.9811
119000.00	0.4812	0.6887	0.8532	0.9651	0.9803	0.9903
136000.00	0.5277	0.7365	0.8884	0.9784	0.9887	0.9950
avg.	0.3290	0.4925	0.6478	0.8011	0.8356	0.8675

--*-* END OF ANALYSIS *-*-*-*

Figure 4.12. SEE1 Analysis for Varying BET1. A PA-28 target aircraft flying at 130 knots approaches at a crossing angle of 90 degrees. The visual range is varied from 2 to 9999 nmi (the latter value representing unlimited visual range). The probability of visual acquisition is evaluated at 12 seconds prior to collision.

5. SEE2 ANALYSIS PROGRAM

5.1 Equivalency Analysis

The visual acquisition model can be used to determine the maximum aircraft speed at which a standard level of visual acquisition performance can be achieved. One use of such an analysis might be to determine the extent to which waivers from normal speed limits are justified in a particular situation. There are two possible approaches to establishing a maximum speed. The first is to impose an absolute requirement on the achieved probability of visual acquisition (PACQ). If this approach is used, then the program SEE1 can be used to determine the maximum speed. A second approach, known as equivalency analysis, requires that the PACQ value be equivalent to that of a reference pair of aircraft flying in the same airspace. The absolute value of PACQ is not fixed. As will be explained below, equivalency analysis has certain advantages. A special program, SEE2, has been written to assist in equivalency analysis.

One difficulty with the use of an absolute PACQ criterion is the need to justify the value of PACQ that is to be required. A reasonable standard would be the value of PACQ that results from compliance with normal federal regulations and practices. However, this value varies greatly with aircraft speeds, approach geometry, and visual range. The PACQ that is attained under worst case legal flight conditions (two aircraft flying at the speed limit, approaching head-on, with barely legal VFR visual range) appears to be too low to serve as a suitable standard. A more reasonable standard could be the typical PACQ value resulting from existing regulations. But the averaging procedure used to find the typical value then becomes an issue. Furthermore, the use of an absolute standard requires that the selected values of all input variables be accurate in an absolute sense. Absolute values may not be known for the traffic environment of interest. (For example, PACQ is affected by the visual range that exists. If the visual range in the airspace where the waiver will apply is uncertain, then the PACQ value that results from a given airspeed will be uncertain.)

When equivalency analysis is applied, the see-and-avoid performance of the subject aircraft against a typical target is compared to the performance of a reference pair of aircraft flying under identical flight conditions. The speed of the subject aircraft is altered until the see-and-avoid performance for the pair involving the subject equals the see-and-avoid performance of the reference pair. The principal output of the program is then the value of the speed required to produce equivalent confidence in visual acquisition.

The advantage of equivalency analysis lies in its insensitivity to the exact values of many input variables. This insensitivity exists because many of the input variables affect the reference pair and the subject pair similarly. (For example, each pair is assumed to be subjected to the same meterological conditions, hence changes in visual range tend to have minimal effect upon the relative performance of the two pairs.)

For SEE2, the nominal reference pair consists of two single-engine general aviation aircraft flying under visual flight rules. SEE2 considers visual acquisition to be successful if either aircraft of a pair acquires successfully.

5.2 Input Data for SEE2 Analysis

Input specification for a SEE2 analysis is more involved than for a SEE1 analysis because two pairs of aircraft must be specified. The input data for SEE2 resides in an input text file. SEE2 reads this file to initialize the analysis. The input file can be altered using an ordinary text editor.

Figure 5.1 is a listing of a typical SEE2 input file. The first line is a file header that can be any string of alphanumeric characters. The second is a title line for the analysis. The third line defines the row variable in terms of its name, index number, minimum, maximum, and step size. The sixth line defines the minimum and maximum speeds that the subject aircraft will be allowed to assume. (If no equivalency can be found within these limits, a failure to converge is declared.) This line also defines the precision (in knots) required to halt equivalency iteration (in this example, 0.01 knots). The next line is a heading line. The next 16 lines contain a table that defines the input variables for all four aircraft (two pairs). The values in the first two columns are redundant, since the reference pair must consist of two identical aircraft.

SEE2 uses the same table of aircraft types as SEE1.

5.3 Computational Structure of SEE2

A flowchart of the basic structure of SEE2 is provided in Fig. 5.2. A listing for SEE2 is provided in Appendix C. Most of the basic SEE1 computational modules described previously in section 4 are employed in SEE2. A row variable is stepped between defined limits to generate a set of speed limits. For each value of the row variable, the probability of acquisition is first computed for the reference pair. It is possible that equivalency cannot be achieved. In such a case, no value is generated for that value of the row variable.

The pair calculation assumes that PACQ for each aircraft is statistically independent. If p_{ij} is the probability of successful acquisition for aircraft i searching for aircraft j, then equivalency exists when

$$(1-p_{12})(1-p_{21}) = (1-p_{34})(1-p_{43})$$

In this equation, the left side represents the probability of acquisition failure for the reference pair. The right side represents the probability of acquisition failure for the subject pair.

```
DATA FILE: g:SEE2.S4 Version c
Sample Analysis
BET1      2       17000.      136000.      17000.
0.    1500.   0.01  { yll, yul   stopitt  }
{VARIABLE     REF AC1    REF AC2    REF AC3    SUB AC4}
BET0           17000      17000      17000      17000
BET1           17000      17000      17000      17000
T1             180.0      180.0      180.0      180.0
T2              12.0       12.0       12.0       12.0
D1               0.0        0.0        0.0        0.0
D2               0.0        0.0        0.0        0.0
R               20.0       20.0       20.0       20.0
DLIM             1.0        1.0        1.0        1.0
VSUB           130.0      130.0      130.0      130.0
VTAR           130.0      130.0      130.0      130.0
IACT               1          1          1          1
XANG           180.0      180.0      180.0      180.0
PL               1.0        1.0        1.0        1.0
PR               0.0        0.0        0.0        0.0
FOVL            -120       -120       -120       -120
FOVR              90         90         90         90
```

Figure 5.1. Input Data File for SEE2.

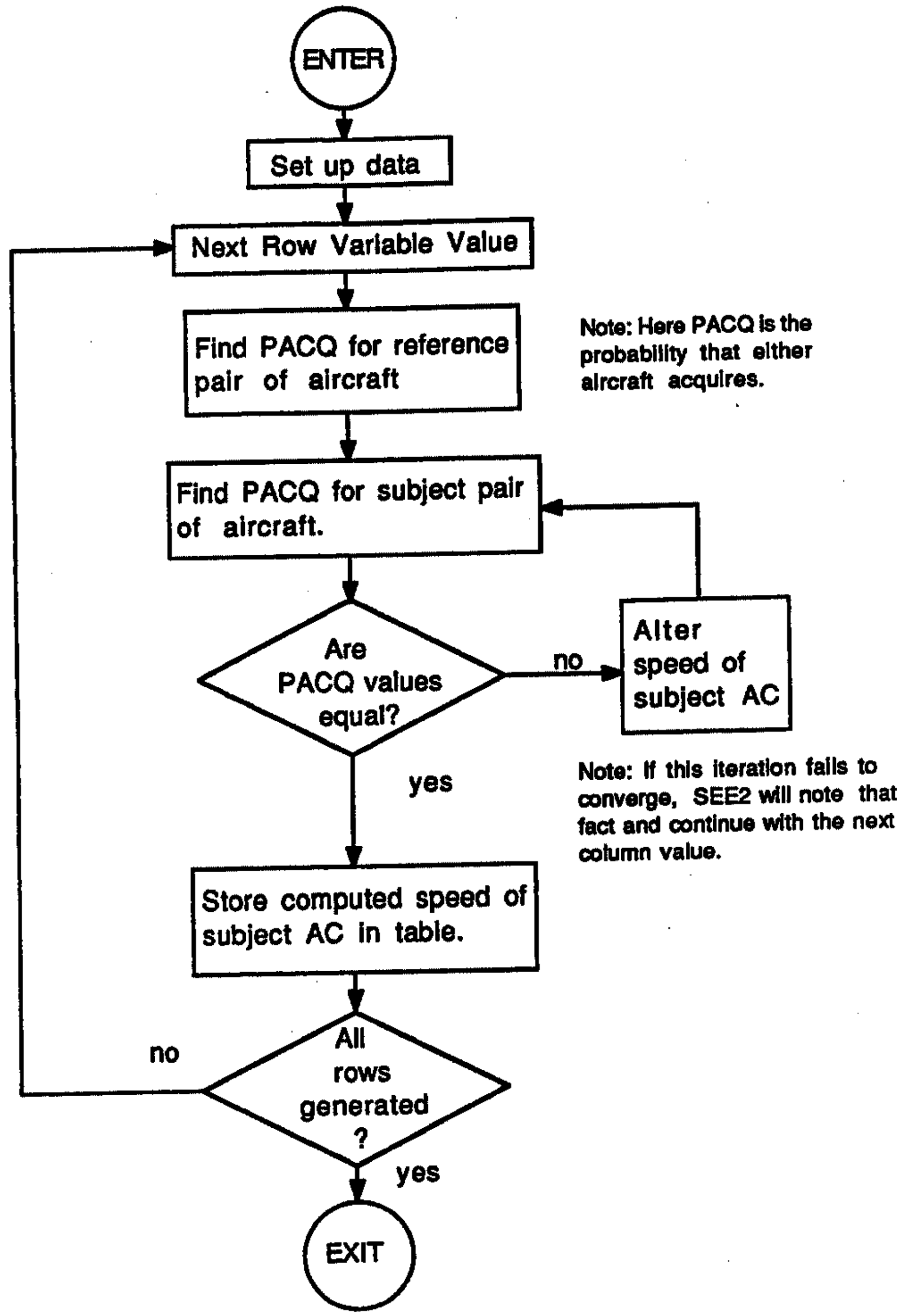

Figure 5.2 Loops controlling output for SEE2.

5.4 Examples of SEE2 Output

Figures 5.3 and 5.4 provide examples of SEE2 output. In Fig. 5.3, the combined visual acquisition probability must be 0.2676. It is seen that increasing BET1 by a factor of 7 (from 17000 to 136000) allows the subject aircraft to increase its speed from 130 kt to 330 kt while maintaining an equivalent level of safety. Note that at 130 knots, each aircraft in the pair has an equal probability of acquiring (p=0.1442) while at 330 kt, the primary responsibility for visual acquisition must rest with the subject (who has probability 0.2418 compared to only 0.03402 for the target).

In Fig. 5.4, a similar analysis is run for a reference crossing angle of 90 degrees. Note that even though the reference acquisition probability has increased by almost a factor of three (from 0.2676 to 0.7622), the equivalent speeds do not change dramatically. This demonstrates the insensitivity of the equivalency analysis to the details of the environmental conditions.

```
*-*-*-* SEE2 VISUAL ACQUISITION ANALYSIS *-*-*-*
        | SEE2 VERSION: G:8/3b/87
        | INPUT DATA SET NAME: G:SEEIN.A00
        | INPUT DATA SET TITLE: Sample Analysis
        | SetFNam=G:SEE2.S4        VFNam =G:SEEVAR.A00
        | ACFNam=G:SEEAC.A00       OutFNam=G:SEE.OUT

INITIAL VALUES OF INPUT VARIABLES:
                    - REFERENCE PAIR -    DUMMY    SUBJECT
                      AC1       AC2        AC3       AC4
      BETO           17000     17000      17000     17000
      BET1           -----     -----      -----     -----    ROW VARIABLE
      T1             180.0     180.0      180.0     180.0
      T2              12.0      12.0       12.0      12.0
      D1              0.00      0.00       0.00      0.00
      D2              0.00      0.00       0.00      0.00
      R               20.0      20.0       20.0      20.0
      DLIM            1.00      1.00       1.00      1.00
      VSUB           -----     -----      -----     -----    FLOATING VARIABLE
      VTAR           130.0     130.0      130.0     130.0
      IACT               1         1          1         1
      XANG           180.0     180.0      180.0     180.0
      PL               1.0       1.0        1.0       1.0
      PR               0.0       0.0        0.0       0.0
      FOVL            -120      -120       -120      -120
      FOVR              90        90         90        90

ROW VARIABLE: BET1 varies from 17000.00 to 136000.0 in steps of 17000.00
FLOATING VARIABLE: VSUB (between limits of   1.00 and   1500.00)

AIRCRAFT TYPES:
                  ACID  TYPE              AX      AY      AZ
             AC1     1  PA28             35.0    85.0    260.0
             AC2     1  PA28             35.0    85.0    260.0
             AC3     1  PA28             35.0    85.0    260.0
             AC4     1  PA28             35.0    85.0    260.0

             ------ OUTPUT TABLE -------
               BET1        VSUB    pacqref   pacq3    pacq4
         <<  17000.000   130.00   0.2676   0.1442   0.1442 >>
         <<  34000.000   175.91   0.2676   0.0986   0.1875 >>
         <<  51000.000   212.11   0.2676   0.0749   0.2083 >>
         <<  68000.000   242.26   0.2676   0.0604   0.2205 >>
         <<  85000.000   268.23   0.2676   0.0506   0.2286 >>
         << 102000.000   291.11   0.2676   0.0435   0.2343 >>
         << 119000.000   311.58   0.2676   0.0382   0.2385 >>
         << 136000.000   330.12   0.2676   0.0340   0.2418 >>

*-*-*-* END OF SEE2 ANALYSIS  *-*-*-*
```

Figure 5.3. SEE2 Output for Data Set Shown in Figure 5.1. The allowable speed of the subject aircraft increases as BET1 for the subject is increased from 17000 to 136000 (a factor of 7).

--*-* SEE2 VISUAL ACQUISITION ANALYSIS *-*-*-*

SEE2 VERSION: G:8/3b/87
INPUT DATA SET NAME: G:SEEIN.A00
INPUT DATA SET TITLE: Sample Analysis
SetFNam=G:SEE2.S4 VFNam =G:SEEVAR.A00
ACFNam=G:SEEAC.A00 OutFNam=G:SEE.OUT

INITIAL VALUES OF INPUT VARIABLES:

	- REFERENCE	PAIR -	DUMMY	SUBJECT	
	AC1	AC2	AC3	AC4	
BETO	17000	17000	17000	17000	
BET1	----	----	----	----	ROW VARIABLE
T1	180.0	180.0	180.0	180.0	
T2	12.0	12.0	12.0	12.0	
D1	0.00	0.00	0.00	0.00	
D2	0.00	0.00	0.00	0.00	
R	20.0	20.0	20.0	20.0	
DLIM	1.00	1.00	1.00	1.00	
VSUB	----	----	----	----	FLOATING VARIABLE
VTAR	130.0	130.0	130.0	130.0	
IACT	1	1	1	1	
XANG	90.0	90.0	90.0	90.0	
PL	1.0	1.0	1.0	1.0	
PR	0.0	0.0	0.0	0.0	
FOVL	-120	-120	-120	-120	
FOVR	90	90	90	90	

ROW VARIABLE: BET1 varies from 17000.00 to 136000.0 in steps of 17000.00
FLOATING VARIABLE: VSUB (between limits of 1.00 and 1500.00)

AIRCRAFT TYPES:

	ACID	TYPE	AX	AY	AZ
AC1	1	PA28	35.0	85.0	260.0
AC2	1	PA28	35.0	85.0	260.0
AC3	1	PA28	35.0	85.0	260.0
AC4	1	PA28	35.0	85.0	260.0

\- - - - - - OUTPUT TABLE - - - - - - -

	BET1	VSUB	pacqref	pacq3	pacq4	
<<	17000.000	130.00	0.7622	0.5124	0.5124	>>
<<	34000.000	179.95	0.7622	0.3299	0.6451	>>
<<	51000.000	219.15	0.7622	0.2281	0.6919	>>
<<	68000.000	251.51	0.7622	0.1685	0.7140	>>
<<	85000.000	279.24	0.7622	0.1308	0.7264	>>
<<	102000.000	303.63	0.7622	0.1053	0.7342	>>
<<	119000.000	325.65	0.7622	0.0884	0.7391	>>
<<	136000.000	345.79	0.7622	0.0771	0.7423	>>

--*-* END OF SEE2 ANALYSIS *-*-*-*

Figure 5.4. SEE2 Output for a Reference Crossing Angle of 90 Degrees. The allowable speed of the subject aircraft increases as BET1 for the subject is increased from 17000 to 136000 (a factor of 7).

6. SELECTION OF INPUT VALUES

This section provides guidance in the selection of input values. For some variables, the proper value will be obvious from the description of the situation of interest. For other variables, the value is not obvious without reference to flight test data or to special tables.

6.1 Search Effectiveness Parameters: BETO and BET1

The search effectiveness parameter, , relates the rate of visual acquisition to the size of the target (see Appendix A for equations). If atmospheric visibility is good, ^{o} is the acquisition rate that is achieved per steradian of subtended target size. For example, if ^{o} is 17,000/steradian-sec and the target subtends 1 micro-steradian, then the acquisition rate is 0.017/sec. and the probability of visual acquisition for each second of search would be 1.7 percent.

^{o} is decreased by pilot workload and is increased by increased vigilance and knowledge of target location (such as that provided by a traffic advisory). Ideally, ^{o} would be derived from flight test data in which the key conditions affecting ^{o} are identical to the situation of interest. However, only a handful of flight test programs to characterize visual acquisition have been conducted. Four flight tests are described below.

6.1.1 Unalerted Search by General Aviation Pilots

In a flight test at M.I.T. Lincoln Laboratory (reference 3), visual acquisition performance was measured for 24 general aviation subject pilots flying a Beech Bonanza on a cross-country course. Concentration upon visual search was avoided by describing the test as a study of differences in VFR pilot techniques under normal flight conditions. Subjects were asked to provide periodic workload ratings, answer formal questions, and to call out all traffic seen. During a 40 minute flight, a second test aircraft made three intercepts, passing 500 feet above or below the subject. The time of each visual acquisition was recorded and the range of each acquisition determined from radar tapes. The experimental technique was judged successful in preventing abnormal pilot emphasis upon the visual search task. The value of ^{o} that resulted was 17,000/ster-sec.

6.1.2 Search by Airline Pilots

In 1957, Wayne Howell of the Civil Aeronautics Administration reported upon a series of flight tests (reference 4) in which subject airline pilots flew a DC-3 aircraft while being intercepted by a second DC-3. A camera was set up to photograph the eye movements of the subject. In one set of flights, the subjects were told that eye movements would be studied, but were not informed that an intercept would be performed. An analysis of the reported data using the visual acquisition model yields a ^{o} value of 34,000/sec. It should be noted that the emphasis upon eye movements may have resulted in greater vigilance on the part of the subjects than would be present in non-test situations. In a second series of flights, pilots were informed that an intercept would be made. This information caused the pilots to alter their scanning technique somewhat, but did not result in a clear improvement in visual acquisition performance. Thus, the β value under both test conditions may correspond to a higher level of search effort than is typical of normal flight.

6.1.3 Alerted Search Using ATARS Traffic Advisories

Flight tests of the Automatic Traffic Advisory and Resolution Service (ATARS) were conducted at the M.I.T. Lincoln Laboratory in 1976-77 (reference 1). General aviation pilots flew a Cessna 172 aircraft during planned intercepts. The traffic advisory display provided intruder bearing to the nearest clock position (30 degree sector). Data from 109 encounters was analyzed and the value of β was determined to be 90,000/ster-sec.

6.1.4 Alerted Search Using TCAS II

In a series of flights at M.I.T. Lincoln Laboratory (reference 2), visual acquisition performance was determined for professional pilots using the Traffic Alert Collision Avoidance System (TCAS II). This system provided a traffic advisory with a bearing accuracy of approximately 8 degrees (one-sigma). The resulting β value was 130,000/ster-sec.

6.1.5 Determining β by Extrapolation

When the conditions of interest do not correspond to any relevant flight test, the value of β must be obtained by extrapolation. Two general rules can be applied. First, β increases in direct proportion to the amount of time devoted to visual search. Thus, when a pilot is alerted to the presence of a target, β can double even if the alert contains no information on the direction from which the target is approaching. Second, β should increase inversely with the angular size of the area that must be searched. Insofar as the search process consists of random sweeps throughout the angular region in which the target may exist, the time required to find the target is proportional to the size of the region. More accurate traffic advisory information will increase β by reducing the angular area to be searched.

6.2 Evaluation Range Parameters: T2 and D2

T2 and D2 determine the modified tau value at which the cumulative probability of visual acquisition (PACQ) is to be evaluated (see description of Module 5 in 4.2.5). Normally D2 is set to zero and T2 represents the lead time at which visual acquisition is required in order for visual avoidance to be effective. In literature dealing with visual collision avoidance, a range of values can be found for the required acquisition lead time. The lowest value appears to be 4.5 seconds for military aircraft (Reference 5). The highest value appears to be 15 seconds for civil transport aircraft using the TCAS collision avoidance system (Reference 2). Military aircraft are expected to have a lower requirement due to the fact that 1) they can accelerate more strongly to avoid collision and 2) military pilots may react faster to a sighting due to air combat training that has involved making quick maneuvers to avoid objects on collision courses. The nominal required lead time used in this handbook are 6 s for military aircraft and 12 s for civil aircraft.

It should be noted that when used as a requirement, T2 is not intended to represent a comfortable lead time, nor to represent the smallest time at which visual acquisition could possibly be of use. Instead, it is intended to represent a point at which the probability of successful visual avoidance has fallen to about 50 percent. This choice allows accurate predictions of the avoidance failure rate since avoidance failures that might occur despite acquisition prior to T2 will be offset by avoidance successes that occur despite acquisition after T2.

6.3 Resolution Limit of Human Eye: DLIM

DLIM is the angular diameter of the smallest high-contrast circular object that can be detected by the pilot's eye. From laboratory tests, a typical value of this resolution limit is 1.0 arc-min. It is unusual for any subject to achieve much better than 0.5 arc-min resolution in the laboratory. In an actual cockpit environment, the effects of vibration, viewing through a window, and so forth may increase DLIM. In flight tests, it has been observed that target aircraft are almost never seen until they exceed 2 arc-min. The nominal value of DLIM used in SEE1 is 1.0 arc-min. The exact value employed is seldom of critical importance, since very little of the total opportunity to acquire accumulates when the target aircraft is near the resolution limit.

REFERENCES

1. Andrews, J.W., "Air-to-air Visual Acquisition Performance with Pilot Warning Instruments (PWI)," Project Report ATC-73, Lincoln Laboratory, M.I.T., (25 April 1977), FAA-RD-77-30.

2. Andrews, J.W., "Air-to-air Visual Acquisition Performance with TCAS II," Project Report ATC-130, Lincoln Laboratory, M.I.T., (27 July 1984), DOT/FAA/PM-84/17.

3. Andrews, J.W., "Air-to-air Visual Acquisition Performance in Unalerted Search," ATC-152, Lincoln Laboratory, M.I.T., (January 1988).

4. Howell, W.D., "Determination of Daytime Conspicuity of Transport Aircraft," CAB-TDR-304, Civil Aeronautics Administration, (May 1957).

5. Wulfeck, J.W., Weisz, A., and Raben, M.W., "Vision in Military Aviation," WADC Technical Report 58-399 (ASTIA Doc. No. AD207780), (November 1958).

APPENDIX A

BASIC EQUATIONS OF THE VISUAL ACQUISITION MODEL

General Formulation of the Model

This appendix provides a concise derivation of the basic equations of the visual acquisition model.

In any visual search process, the instantaneous acquisition rate can be defined as the probability of visual acquisition per instant of time, i.e.,

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P\,[\text{acq in } \Delta t \text{ at time } t]}{\Delta t} \tag{1}$$

The classic Poisson process models a situation in which the probability of an event occurring in each instant of time is constant. In equation (1), the acquisition rate can change with time. The cumulative probability of acquisition is then modeled as a non-homogenous Poisson process. The cumulative probability of acquisition by time t_2 can be written

$$P\,[\text{acq by } t_2] = 1.0 - \exp\left[-\int_{-\infty}^{t_2} \lambda(t)\;dt\right] \tag{2}$$

TABLE A.1 NOTATION

A	visual area presented by the target aircraft
$C(r)$	target/background contrast at range r
C_0	inherent contrast of target with background
t	time
r	range (separation between aircraft)
r_1	range at which search begins
r_2	range at which search terminates
R	visual range (point at which contrast degrades by 95%)
rdot	range rate
β	search effectiveness parameter

The above equation is a purely mathematical consequence of the way in which visual acquisition is described. The usefulness of the model hinges upon our ability to define the acquisition rate λ for specified search conditions. In examining several sets of experimental data, it was determined that λ at any instant of time is proportional to the solid angle subtended by the target, i.e.

$$\lambda = \beta \frac{A}{r^2} \tag{3}$$

It is known from laboratory experiments that it is actually the product of the target area and target contrast that determine the detectability of a target. Hence, if contrast varies, the effect can be modeled by substituting the area-contrast product for the area above:

$$= \frac{C(r)}{C_0} \frac{A}{r^2} \tag{4}$$

For vision through a homogenous atmosphere, contrast decreases exponentially with range according to Koschmieder's law:

$$C(r) = C_0 \exp \left[\frac{-2.996\ r}{R} \right] \tag{5}$$

Thus the acquisition rate for search through a homogenous atmosphere can be written

$$\lambda = \beta \frac{A}{r^2} \exp \left[\frac{-2.996\ r}{R} \right] \tag{6}$$

The cumulative probability of acquisition by time t_2 is obtained through integration of λ according to equation (2):

$$P[\text{acq by } t2] = 1.0 - \exp \left[-\int_{\infty}^{t_2} \frac{\beta A}{r^2} \exp \left(\frac{-2.996\ r}{R} \right) dt \right] \tag{7}$$

This equation is the most general result of the visual acquisition model since it allows β, A, and r to vary with time in an arbitrary manner.

Application to a Nominal Collision Course

For an aircraft on an unaccelerated collision course, it can often be assumed that A is constant and that the range is decreasing at a constant rate. For any such period in which β is also constant, the opportunity for acquisition is described by the integral of the size-contrast product, Q :

$$Q(t) = A \int_{-\infty}^{t} \frac{1}{r^2} \operatorname{Exp}\left[\frac{-2.996\, r}{R}\right] dt \qquad (8)$$

The cumulative probability of visual acquisition is

$$P\,[\text{acq by } t] = 1 - \exp\,[-\beta\, Q(t)] \qquad (9)$$

The nominal units of Q are steradians-sec. One unit can accumulate from a target subtending one thousandth of a steradian being in the field of view for one thousand seconds. In practice, most targets of interest will subtend only a few millionths of a steradian.

The opportunity that accumulates as the target moves from r_1 to r_2 ($r_2 < r_1$) can be written

$$Q(t) = \frac{A}{|\text{rdot}|}\left[\frac{E_2(2.996 r_2/R]}{r_2} - \frac{E_2(2.996 r_1/R]}{r_1}\right] \qquad (10)$$

where E_2 is the second-order exponential integral

$$E_2(z) = \int_{1}^{\infty} \frac{\operatorname{Exp}(-zy)}{y^2}$$

This integral can be evaluated by a series expansion (see Pascal function E2 in Appendix C).

Expressions for Infinite Visual Range

When the visual range is infinite (perfectly clear atmosphere), the exponential integrals in (10) have a value of unity regardless of the range at which they are evaluated. The expression for probability of acquisition simplifies to

$$P[\text{acq by } r_2] = 1 - \exp\left[\frac{-\beta A}{|\text{rdot}|}(1/r_2 - 1/r_1)\right] \qquad (11)$$

The probability density function for the range at which acquisition occurs is then

$$fracq(r) = \frac{\beta A}{|rdot|\ r^2} \exp\left[\frac{-\beta A}{|rdot|}(1/r - 1/r_1)\right] \qquad 0 < r < r_1 \qquad (12)$$

$$= 0 \qquad \text{elsewhere}$$

The mean range of visual acquisition computed from the above pdf is

$$E[racq] = q\ r_1 \exp[q]\ \left[-0.57721 + Ln(1/q) - \sum_{n=1}^{\infty} \frac{(-1)^n q^n}{n\ n!}\right] \qquad (13)$$

where $q = \beta A/(|rdot|*r_1|$

The median range of acquisition is the range at which there is a 50 percent probability that visual acquisition has occurred. It can be written:

$$r_{0.5} = \frac{1}{|rdot|(Ln\ 2)/(\beta A) + 1/r_1} \qquad (14)$$

APPENDIX B

FILE STRUCTURE FOR SEE PROGRAMS

The SEE programs are written in the Pascal computer language using the Turbo Pascal compiler from Borland International. This compiler is available for a variety of microcomputer operating systems, including CP/M and MSDOS. A CP/M version is described below.

Table B.1 shows the files that exist on disk for the CP/M version of SEE1. All text files are indicated by "Y" in the second column of the table. Text files can be modified by using the "non-document" or "data" mode of any text editor (Caution: Do not alter these files using the word processing or "document" mode of a word processing program. Such a mode inserts control codes such as page breaks and soft hyphens into the data. The SEE programs will not function if they encounter such control codes in a data file).

TABLE B.1

FILES FOUND ON THE CP/M DISK

FILE NAME	TEXT	DESCRIPTION
SEEAC.B03	Y	Table of aircraft types (names and principal areas).
SEEMODS.B03	Y	Common procedures and functions for SEE1 and SEE2, including EXPP,E2,GETVAR, MODULE1, MODULE2, MODULE3, MODULE4, MODULE5, MODULE6, MODULE7, READR and SGN.
SEEVAR.B03	Y	Initialization data (variable names, nominal values, and conversion factors between internal and external units).
SEE1SET.B03	Y	Set-up file for SEE1. Contains names of all other files used by SEE1. Also contains printer set-up string.
SEE2IN.B03	Y	Simulation initialization data for SEE2. Contains definition of reference and subject aircraft pairs.
SEE2SET.B03	Y	Set-up file for SEE2. Contains names of all other files used by SEE2. Also contains printer set-up string.
SEE1.CMD	N	CP/M command file for SEE1 (compiled version of SEE1).
SEE1.B03	Y	SEE1 source code (Turbo Pascal)
SEE2.CMD	N	CP/M command file for SEE2 (compiled version of SEE2).
SEE2.B03	Y	SEE2 source code (Turbo Pascal).

A basic set-up file, SEE1SET.B03, is used to specify file names for all other files used during the running of SEE1. This file also specifies the printer initialization string to be sent to the printer prior to generation of printed output. Figure B-1 contains the listing for the version of SEE1SET.B03 provided on the start-up disk. The corresponding file for SEE2 is shown in Fig. B-2.

Aircraft data (names and principal areas) are contained in file SEEAC.B03. Figure B-3 contains a listing for the default aircraft file provided on the start-up disk.

Disk file SEEVAR.B03 contains the names of the input/output variables, their nominal values, and the conversion factors used to convert from external to internal units. Internal to SEE1, all units are based upon nautical miles, seconds, and radians. Figure B-4 provides a listing of the default file.

```
SEE1 SET-UP FILE : Version B03
A:SEEVAR.B03         {VARIABLE NAME FILE}
A:SEEAC.B03          {AIRCRAFT TYPE FILE}
A:SEE1.OUT            {OUTPUT DATA FILE}
27  91  50  119  -1  -1       {SIX ASCII CHAR FOR PRINTER SET-UP: -1 TO SKIP}
```

Figure B-1 Input data file SEE1SET.B03

```
SEE2 SET-UP  FILE Version B03
A:SEE2IN.B03             {SET-UP DATA FILE}
A:SEEVAR.B03          {VARIABLE NAME FILE}
A:SEEAC.B03           {AIRCRAFT TYPE FILE}
A:SEE2.OUT             {OUTPUT DATA FILE}
27  91  50  119  -1  -1       {SIX ASCII CHAR FOR PRINTER SET-UP: -1 TO SKIP}
```

Figure B-2 Input data file SEE2SET.B03

```
1 PA28          35.    85.    260.
2 B727         400.  1900.   3100.
3 B747        1200.  5700.   9300.
4 DC-9         300.  1425.   2325.
5 C421          81.   171.    417.
6 U-21         127.   267.    652.
7 F18           50.   280.    540.
```

Figure B-3 Aircraft data file SEEAC.B03

NAME	NOMINAL	CONV	EXT UNITS
BETO	17000.0	1.0	/STER-SEC
BET1	17000.0	1.0	/STER-SEC
T1	180.0	1.0	SEC
T2	12.0	1.0	SEC
D1	0.0	1.0	NMI
D2	0.0	1.0	NMI
R	20.0	1.0	NMI
DLIM	1.0	2.90883E-04	ARC-MIN
VSUB	180.0	2.7777E-04	KT
VSUB	130.0	2.7777E-04	KT
IACT	1.0	1.0	AC TYPE
XANG	180.0	0.017453	DEG
PL	1.0	1.0	LEFT PILOT
PR	0.0	1.0	RIGHT PILOT
FOVL	-120.0	0.017453	DEG
FOVR	90.0	0.017453	DEG
RDOT	300.0	2.7777E-04	KT
AX	35.0	2.70872E-08	SQ. FT.
AY	85.0	2.70872E-08	SQ. FT.
AZ	260.0	2.70872E-08	SQ. FT.
AREA	35.0	2.70872E-08	SQ. FT.
THT	0.0	0.017453	DEG
THS	0.0	0.017453	DEG
RNG1	3.78	1.00	NMI
RNG2	1.27	1.00	NMI
BEFO	17000.	1.00	/STER-SEC
BEF1	17000.	1.00	/STER-SEC
PACQ	0.	1.00	-

Figure B-4 Variable initialization file SEEVAR.B03

APPENDIX C

SEE LISTINGS

This appendix provides listings for the SEE1 programs. The programs were written in the Pascal computer language and compiled using the Turbo Pascal compiler from Borland International, Incorporated. Section C.1 contains all procedures, functions, and main program code for SEE1, arranged in alphabetical order of name (the main program is listed as SEE1). Section C.2 contains all code for SEE2 with the exception of routines that have been previously listed in Section C.1. It should be noted that in Turbo Pascal, if a procedure B calls procedure A, then A must be listed first in the program listing. Hence the actual source code that is compiled will not have procedures in alphabetical order. The source code also uses an "include" directive to merge the common procedures in file MODULES with the main program listing during compilation.

C.1 Listing for SEE1

```
PROCEDURE Dispp(Var pp,ppe,conv : ppty; name : namety;
 index,jndex : integer);
{DISPLAY INPUT VARIABLES TO CONSOLE}
VAR skip : string[12];  i,j,ii : integer;
begin
LowVideo;
Exunits(pp,ppe,conv);
For i:=1 to 6 do
   begin
   For j:=1 to 3 do
      begin
      ii:=6*(j-1)+i;
      If (ii<=16) then
         begin
         Write('      ',name[ii],'  ');
         If (ppe[ii]<10.0) then Write(ppe[ii]:10:2) else Write(ppe[ii]:10:1);
         If (ii=index) or (ii=jndex) then Write('*') else Write(' ');
         end;
      end;
   Writeln;
   end;
Writeln;
Writeln('                * = value on first iteration');
NormVideo;
end;      {of PROCEDURE Dispp}

FUNCTION Expp(x : real) : real;
begin  {Exponentiation with overflow/underflow protection}
If (x>85.0) then Expp:=1.0e+37
else If (x<-85.0) then Expp:=1.0e-37
else  Expp:=exp(x);
end;
```

```
FUNCTION E2(x : real) : real;
{Evaluate exponential integral of order 2 from x to infinity}
Var
     df,f1,f2 : real;      ii,it : integer;
begin
e2:=1.0E-37;
If (x<9.0) then
     begin
     f1:=expp(-x)+0.57721*x+x*Ln(x);
     df:=-x*x;
     f2:=df;
     it:=0;
     Repeat
          it:=it+1;  ii:=(it+1)*(it+1);
          df:=-it*x*df/ii;
          f2:=f2+df;
     Until (abs(df)<1.0E-09);
     e2:=f1+f2;
     end;
end;

PROCEDURE Exunits(Var pp,ppe,conv : ppty);
{LOAD PPE WITH VARIABLES  IN EXTERNAL UNITS}
Var i : integer;
begin
For i:=1 to 28 do ppe[i]:=pp[i]/conv[i];
end;     {of Exunits}

FUNCTION Fpacq(Var pp : ppty; Var tabl : tabty) : real;
Var fov : real;
begin         {COMPUTE PROB OF VISUAL FOR CONDITIONS IN PP}
Module1(pp[11],tabl,pp[18],pp[19],pp[20]);
Module2(pp[9],pp[10],pp[12],pp[17],pp[22],pp[23]);
Module3(pp[18],pp[19],pp[23],pp[21]);
Module4(pp[1],pp[2],pp[15],pp[16],pp[13],pp[14],pp[22],pp[26],pp[27],fov);
Module5(pp[4],pp[6],pp[17],pp[25]);
Module6(pp[3],pp[5],pp[17],pp[24]);
IF (DEBUG>3) THEN BEGIN WRITELN(Lst,'INPUT TO MODULE 7:  AREA=',pp[21]:10:8,
' BEFO,BEF1=',pp[26]:9:0,pp[27]:9:0);
WRITELN(Lst,'        DLIM=',pp[8]:10:8,' R=',pp[7]:7:3,'  RDOT=',pp[17]:9:5);
WRITELN(Lst,'        RNG1=',pp[24]:7:3,'  RNG2=',pp[25]:7:3);  END;
Module7(pp[21],pp[26],pp[27],pp[8],pp[7],pp[17],pp[24],pp[25],pp[28]);
Fpacq:=pp[28];
end;      {OF FUNCTION FPACQ}

PROCEDURE GetVar(Var st : strng4; Var index : integer);
{VALIDATE VARIABLE NAME AND GET INDEX. CR RETURNS INDEX=-1}
Var i : integer;
begin
Repeat
```

```
Write('>>'); Readln(st);  index:=0;
For i:=1 to Length(st) do st[i]:=UpCase(st[i]);
While (Length(st)<4) do st:=st+' ';
If (st<>'    ') then
     begin
     index:=-1;
     For i:=1 to 16  do If (name[i]=st) then  index:=i;
     IF (index<0) then Writeln(Chr(7),'ILLEGAL NAME - TRY AGAIN')
     end;
Until (index>=0);
end;  { OF GETVAR MODIFIED 12/30/87}

PROCEDURE Lstpp(Var pp,ppe,conv : ppty; Var name : namety);
{PRINT INPUT VARIABLES TO PRINTER}
Var i,ii,j : integer;
begin
Exunits(pp,ppe,conv);
For i:=1 to 6 do
   begin
   For j:=1 to 3 do
      begin
      ii:=6*(j-1)+i;
      If (ii<=16) then
         begin
         Write(OutF,'          ',name[ii],'  ');
         If (ppe[ii]<10.0) then Write(OutF,ppe[ii]:10:2) else Write(OutF,ppe[ii]:10:1);
         If (ii=index) or (ii=jndex) then Write(OutF,'*') else Write(OutF,' ');
         end;
      end;
   Writeln(OutF);
   end;
Writeln(OutF);
Writeln(OutF,'                          * = value on first iteration');
Writeln(OutF);
end;       {of PROCEDURE Lstpp}

PROCEDURE Module1(xiact : real; tabl : tabty; Var ax,ay,az : real);
Var iact : integer;
{ASSIGN TARGET AIRCRAFT AREAS BY INDEXING TAB1}
begin
iact:=Round(xiact);
ax:=tabl[iact,1]*sqnmi;
ay:=tabl[iact,2]*sqnmi;
az:=tabl[iact,3]*sqnmi;
end;        {of Procedure Module1  7/30/87}

PROCEDURE Module2(Var v1,v2,xang,rdot,th1,th2 : real);
{COMPUTE TH1,TH2 AND RDOT ASSUMNG COLLISION COURSE}
Var
     dif,cosmax,cosx,sinth,sinx,vcosmax,vrat : real;
```

```
begin
If (Abs(v2-v1)<0.00001)  then     {USE EQUAL SPEED FORMULAS}
     begin
     If (xang=0.0) then xang:=0.00001;  {DISALLOW DEGENERATE CASE}
     th1:=xang/2-0.5*pi*sgn(xang);
     th2:=-th1;
     rdot:=-2*v1*abs(sin(xang/2));
     end
else                             {USE UNEQUAL SPEED FORMULAS}
     begin
     vrat:=v2/v1; cosx:=cos(xang);
     If (xang<>0.0) then
        begin
        th1:=arctan(-vrat*sin(xang)/(1.0-vrat*cosx));
        If (cosx>1.0/vrat) then th1:=th1-pi*sgn(th1);
        end
     else if (vrat<1.0) then th1:=0.0
     else th1:=pi;
     th2:=pi+th1-xang;
     If (abs(th2)>pi) then th2:=th2-2*pi*sgn(th2);
     rdot:=-v1*cos(th1)-v2*cos(th2);
     end;
end;                 {of Module 2   Version 3/11/87}


PROCEDURE Module3(ax,ay,tht : real; Var area : real);
{COMPUTE VISUAL AREA OF TARGET. USE APPROXIMATE CORRECTION FOR
HIDDEN AREAS}
Var
    axx,ayy : real;
begin
axx:=ax*abs(cos(tht));
ayy:=ay*abs(sin(tht));
If (axx>ayy) then area:=axx+ayy/3  else area:=ayy+axx/3;
end;


PROCEDURE Module4(bet0,bet1,fovl,fovr,pl,pr,th1 : real;
Var bef0,bef1,fov : real);
{DETERMINE EFFECTIVE VALUES OF BETA BEFORE AND AFTER TRANSITION}
{FOV=0 IF TARGET NOT IN FIELD OF VIEW OF ANY PILOT}
begin
bef0:=0.0; bef1:=0.0; fov:=0.0;
{IF IN LEFT FOV, ADD CONTRIBUTION OF PILOT IN LEFT SEAT}
If (th1>=fovl) and (th1<=fovr) then
     begin
     bef0:=bef0+pl*bet0;
```

```
        bef1:=bef1+pl*bet1;
        fov:=fov+pl;
        end;
   {IF IN RIGHT FOV, ADD CONTRIBUTION OF PILOT IN RIGHT SEAT}
If (th1>=-fovr) and (th1<=-fovl) then
     begin
     bef0:=bef0+pr*bet0;
     bef1:=bef1+pr*bet1;
     fov:=fov+pr;
     end;
end;      {of Module4 12/29/87}


PROCEDURE Module5(t2,d2,rdot : real; Var rng2 : real);
{USE MODIFIED TAU TO COMPUTE RANGE AT WHICH PACQ IS EVALUATED}
begin
rng2:=d2-rdot*t2;
end;


PROCEDURE Module6(t1,d1,rdot : real; Var rng1 : real);
{USE MODIFIED TAU TO COMPUTE RANGE AT WHICH BETA TRANSITIONS}
begin
rng1:=d1-rdot*t1;
end; {OF MODULE6}


PROCEDURE Module7(area,bef0,bef1,dlim,r,rdot,rng1,rng2 : real; Var pacq : real);
{COMPUTE PACQ}
Var
     absrd,pacq0,rlim,xlim,xpost,xprior,xx,xx1,xx2 : real;
begin
absrd:=abs(rdot)+0.00001;
pacq0:=0.0;  pacq:=0.0;
rlim:=1.1284*sqrt(area)/dlim;  {RLIM CANNOT EXCEED RESOLUTION LIMIT OF EYE}
If (rlim>r) then rlim:=r;     {RLIM CANNOT EXCEED VISUAL RANGE}
{- - - COMPUTE INTEGRAL OVER THREE POSSIBLE TIME INTERVALS}
xlim:=e2(2.996*rlim/r)/rlim;
xx1:=e2(2.996*rng1/r)/rng1;
xx2:=e2(2.996*rng2/r)/rng2;
If (rlim>rng1) then
     begin                     {ADD INTEGRAL PRIOR TO TRANSITION}
     If (rng1>rng2) then xx:=xx+bef0*(xx1-xlim)
     else   If (rng2<rlim) then xx:=xx+bef0*(xx2-xlim);
     end;
```

```
IF (debug>4) then
   begin
   WRITELN(Lst,' IN MODULE7: RLIM,RNG1,RNG2=',RLIM:8:4,RNG1:8:4,RNG2:8:4);
   WRITELN(Lst,'             XLIM,XX1,XX2=',XLIM:8:4,XX1:8:4,XX2:8:4);
   WRITELN(Lst,'             XX AT BEFO=',XX:10:4);
   end;
If (rng1>rng2) then
     begin                    {ADD INTEGRAL AFTER TRANSITION}
     If  (rlim>rng1) then xx:=xx+bef1*(xx2-xx1)
     else If (rlim>rng2) then xx:=xx+bef1*(xx2-xlim);
     end;
xx:=area*xx/absrd;
pacq:=1.0-expp(-xx);
end;  {OF MODULE 7}

PROCEDURE Out_const(Var pp,ppe,conv : ppty; Var tabl : tabty;
Var acname : acnamety; Var index,jndex : integer);
{- - - PRINT OUT COMPUTED VARIABLES THAT ARE UNAFFECTED BY ITERATION  - - - -}
Var  con11,con17 : Boolean;
begin
Module1(pp[11],tabl,pp[18],pp[19],pp[20]);
Module2(pp[9],pp[10],pp[12],pp[17],pp[22],pp[23]);
Module3(pp[18],pp[19],pp[23],pp[21]);
Exunits(pp,ppe,conv);
If (index=11) or (jndex=11) then con11:=FALSE else con11:=TRUE;
con17:=TRUE;
If (index=9) or (jndex=9) or (index=10) or (jndex=10) then con17:=FALSE;
If (index=12) or (jndex=12) then con17:=FALSE;
If con11 then
   begin
   Writeln(OutF,'     | Target aircraft is ',acname[iact]);
   Write(OutF,'     | AX=',ppe[18]:6:1,' sq ft   AY=',ppe[19]:6:1,
   ' sq ft   AZ=',ppe[20]:6:1,' sq ft');
   If con17 then Writeln(OutF,'   AREA=',ppe[21]:7:1,' sq ft')
   else Writeln(OutF);
   end;
If con17 then
   begin
   Write(OutF,'     | RDOT=',ppe[17]:7:1,' kt    THT=',ppe[22]:7:1,' deg');
   Write(OutF,'    THS=',ppe[23]:7:1,' deg');
   end;
Writeln(OutF);
end;  {OF PROCEDURE Out_const }
PROCEDURE Printpp(Var pp,ppe,conv : ppty; name : namety);
{PRINT-OUT VARIABLES IN EXTERNAL UNITS}
Var   i,ii,j : integer;
begin
Exunits(pp,ppe,conv);
ii:=0;
```

```
For i:=1 to 27 do
     begin
          If (i<28) then
               begin
               Write(OutF,name[i],ppe[i]:10:2,'  ');
               ii:=ii+1;
               If (ii=5) then begin Writeln(OutF); ii:=0; end;
               end;
     end;
     Writeln(OutF,name[28],ppe[28]:10:4);
end;     {of PROCEDURE Printpp}

PROCEDURE  ReadR(Var nitems : integer; Var buf : rbuff);
LABEL RETYPE;
Var i,len,valid : integer; ch : char; s : string[255];  r : string[32];
begin
RETYPE: nitems:=0;  r:='';
Readln(s);  s:=s+','; len:=Length(s);
For i:=1 to len do
   begin
   ch:=s[i];
   If (ch<>' ') and (ch<>',') then r:=r+ch
   else
      begin
      If (Length(r)>0) then
         begin
         nitems:=nitems+1;
         Val(r,buf[nitems],valid);
         If (valid<>0) then
            begin
            Writeln(Chr(7),' Error at "',r,'" (character ',valid:3,')');
            Write('Retype line');
            goto RETYPE;
            end;
         r:='';
         end;
      end;
   end;
end;  {of PROCEDURE ReadR Revised 1/10a/86}

Program SEE1;
{$R+}
Const
   version='B03'; sqnmi=2.70872E-08;  nmisec=2.77777E-04;
   lm='      '; nrowsmax=128;
Label  L1,REDEF,START;
Type
     namety = array[1..32] of string[4];
     acnamety = array[1..24] of string[12];
```

```
     ppty = array[1..32] of real;
     rbuff = array[1..32] of real;
     strng4 = string[4];
     tabty = array[1..16,1..4] of real;
Var
     InF,OutF,TempF : text;
     fov,x,xmin,xmax,xstep,xval : real;
     debug,i,iact,icol,index,ix,j,jmax,jndex : integer;
     long,menu,nacty,nrows,nitems : integer;
     conv,nomval,pp,ppe : ppty;       buf : rbuff;
     ibuff : array[1..32] of integer;
     table : array[1..nrowsmax,1..8] of real;
     name : namety;    st : string[4];     note,s,ss : string[255];
     acname : acnamety;      ufov : Boolean;
     pavg,ppcol : array[1..8] of real;
     tabl : tabty;
     ACFNam,InFNam,OutFNam,printer,VFNam : string[14];
{$I SEEMODS.B03}
PROCEDURE Dispp(Var pp,ppe,conv : ppty; name : namety;
 index,jndex : integer);
{DISPLAY INPUT VARIABLES TO CONSOLE}
VAR skip : string[12];  i,j,ii : integer;
begin
LowVideo;
Exunits(pp,ppe,conv);
For i:=1 to 6 do
   begin
   For j:=1 to 3 do
      begin
      ii:=6*(j-1)+i;
      If (ii<=16) then
         begin
         Write('      ',name[ii],'  ');
         If (ppe[ii]<10.0) then Write(ppe[ii]:10:2) else Write(ppe[ii]:10:1);
         If (ii=index) or (ii=jndex) then Write('*') else Write(' ');
         end;
      end;
   Writeln;
   end;
Writeln;
Writeln('               * = value on first iteration');
NormVideo;
end;      {of PROCEDURE Dispp}
PROCEDURE Lstpp(Var pp,ppe,conv : ppty; Var name : namety);
{PRINT INPUT VARIABLES TO PRINTER}
Var i,ii,j : integer;
begin
Exunits(pp,ppe,conv);
For i:=1 to 6 do
   begin
   For j:=1 to 3 do
```

```
      begin
      ii:=6*(j-1)+i;
      If (ii<=16) then
         begin
         Write(OutF,'          ',name[ii],'  ');
         If (ppe[ii]<10.0) then Write(OutF,ppe[ii]:10:2) else Write(OutF,ppe[ii]:10:1);
         If (ii=index) or (ii=jndex) then Write(OutF,'*') else Write(OutF,' ');
         end;
      end;
   Writeln(OutF);
   end;
Writeln(OutF);
Writeln(OutF,'                         * = value on first iteration');
Writeln(OutF);
end;       {of PROCEDURE Lstpp}
PROCEDURE Out_const(Var pp,ppe,conv : ppty; Var tabl : tabty;
Var acname : acnamety; Var index,jndex : integer);
{- - - PRINT OUT COMPUTED VARIABLES THAT ARE UNAFFECTED BY ITERATION  - - - -}
Var  con11,con17 : Boolean;
begin
Module1(pp[11],tabl,pp[18],pp[19],pp[20]);
Module2(pp[9],pp[10],pp[12],pp[17],pp[22],pp[23]);
Module3(pp[18],pp[19],pp[23],pp[21]);
Exunits(pp,ppe,conv);
If (index=11) or (jndex=11) then con11:=FALSE else con11:=TRUE;
con17:=TRUE;
If (index=9) or (jndex=9) or (index=10) or (jndex=10) then con17:=FALSE;
If (index=12) or (jndex=12) then con17:=FALSE;
If con11 then
   begin
   Writeln(OutF,'     | Target aircraft is ',acname[iact]);
   Write(OutF,'     | AX=',ppe[18]:6:1,' sq ft   AY=',ppe[19]:6:1,
   ' sq ft   AZ=',ppe[20]:6:1,' sq ft');
   If con17 then Writeln(OutF,'   AREA=',ppe[21]:7:1,' sq ft')
   else Writeln(OutF);
   end;
If con17 then
   begin
   Write(OutF,'     | RDOT=',ppe[17]:7:1,' kt    THT=',ppe[22]:7:1,' deg');
   Write(OutF,'    THS=',ppe[23]:7:1,' deg');
   end;
Writeln(OutF);
end;
PROCEDURE Printpp(Var pp,ppe,conv : ppty; name : namety);
{PRINT-OUT VARIABLES IN EXTERNAL UNITS}
Var   i,ii,j : integer;
begin
Exunits(pp,ppe,conv);
ii:=0;
For i:=1 to 27 do
```

```
     begin
          If (i<28) then
               begin
               Write(OutF,name[i],ppe[i]:10:2,'   ');
               ii:=ii+1;
               If (ii=5) then begin Writeln(OutF); ii:=0; end;
               end;
     end;
     Writeln(OutF,name[28],ppe[28]:10:4);
end;     {of PROCEDURE Printpp}
begin  {- - - -  - - - -  - - - - - - - - - - - - BEGIN SEE1 MAIN PROGRAM - - }
debug:=0;
ClrScr;  GotoXY(15,1);
Write('SEE1 VISUAL ACQUISITION ANALYSIS - Version ',version);
GotoXY(30,5); Write('written for ');  GotoXY(19,7);
Write('The Federal Aviation Administration');
GotoXY(30,8); Write('(AT-240)');  GotoXY(34,10); Write('by');
GotoXY(23,12); Write('M.I.T. Lincoln Laboratory'); GotoXY(17,14);
Write('244 Wood St., Lexington, MA 02173-0073'); GotoXY(1,23);
Write('CR to continue>>'); Readln(st);
If (st='menu') then
   begin
   Write('Type debug>>'); Readln(debug);
   end;
InFNam:='SEE1SET.B03'; Writeln('      | INIT FILE NAME: ',InFNam);
Assign(InF,InFNam); Reset(InF);
Readln(InF,s); Writeln('      | INIT FILE TITLE:',s);
Readln(InF,VFNam); Readln(InF,ACFNam); Readln(InF,OutFNam);
{ - - - READ PRINTER INITIALIZATION STRING AND INITIALIZE PRINTER }
Readln(InF,ibuff[1],ibuff[2],ibuff[3],ibuff[4],ibuff[5],ibuff[6]);
printer:=''; For i:=1 to 6 do
     If (ibuff[i]>0) then printer:=printer+Chr(ibuff[i]);
If (printer<>'') then Write(Lst,printer);
{ - - - READ TABLE OF AIRCRAFT CHARACTERISTICS - - - - - - }
Assign(TempF,ACFNam);  Reset(TempF);
nacty:=0;
While not Eof(TempF) do
begin
nacty:=nacty+1;
Readln(TempF,i,acname[nacty],tabl[nacty,1],tabl[nacty,2],tabl[nacty,3]);
end;
Close(TempF);
{ - - - - READ IN VARIABLE NAMES AND NOMINAL VALUES - - - }
Assign(TempF,VFNam); Reset(TempF);
Readln(TempF);    {Skip label line}
For i:=1 to 28 do
Readln(TempF,name[i],nomval[i],conv[i]);
Close(TempF);
Write('Type 1 for long output (CR for summary table only)>>');
long:=0; Readln(long);
```

```
START:
For i:=1 to 28 do
     begin
     ppe[i]:=nomval[i];
     pp[i]:=nomval[i]*conv[i];    {Initialize variables to nominal values}
     end;
{ - - - - - - - - - - - - - - - - - - - - - - - - SET INPUT CONDITIONS  - - - - }
REDEF :
Repeat
Write('Type name of row variable '); GetVar(st,index);
Until (index>0);
Write('Type xmin,xmax,xstep for ',name[index],'>>');
ReadR(nitems,buf); xmin:=buf[1]; xmax:=buf[2]; xstep:=buf[3];
nrows:=Round((xmax-xmin)/xstep+0.5);
If (nrows>nrowsmax) or (nrows<=0) then
   begin
   Writeln(Chr(7),'>>>> NO. OF ROWS MUST BE BETWEEN 1 AND ',nrowsmax:5);
   Writeln('>>>> PLEASE INPUT ROW DATA AGAIN.');
   Goto REDEF;
   end;
pp[index]:=xmin*conv[index];
Repeat
Write('Type name of column variable '); GetVar(st,jndex);
Until (jndex>0);
Writeln('Type up to 8 values for ', name[jndex],' >>');
Readr(jmax,buf);  For j:=1 to jmax do ppcol[j]:=buf[j];
pp[jndex]:=ppcol[1]*conv[jndex];
{- -  - - - - - - - - - - SELECT TARGET AIRCRAFT TYPE - - - - - - - - }
Writeln('TABLE OF DEFINED TARGET AIRCRAFT TYPES :');
Writeln('      IACT  NAME                AX       AY      AZ');
For i:=1 to nacty do
WRITELN('     ',i:3,'  ',acname[i],'  ',tabl[i,1]:6:0,tabl[i,2]:8:0,
tabl[i,3]:8:0);
Write('Type IACT for target aircraft (CR to define new type)>>');
iact:=-1; Readln(iact);  pp[11]:=1.0*iact;
If (iact<=0) then
     begin
     nacty:=nacty+1; iact:=nacty;  pp[11]:=1.0*nacty;
     Write('Type symbolic name for type (12 char max>>');
     Readln(acname[iact]);
     While (Length(acname[iact])<12) do acname[iact]:=acname[iact]+' ';
     Writeln('Type AX,AY,AZ (in sq ft)>>');
     ReadR(nitems,buf);
     tabl[iact,1]:=buf[1]; tabl[iact,2]:=buf[2]; tabl[iact,3]:=buf[3];
     end;
L1 :
{ - - - - - - - - - - - - - - - - - DISPLAY SET-UP TO CONSOLE FOR APPROVAL - - - }
ClrScr;  GotoXY(10,1); Write('*-*-*-*-*-*-* SEE1 ANALYSIS - Version ',version);
```

```
Writeln(' *-*-*-*-*-*-*-*');
Writeln('ROW VARIABLE :   '); LowVideo;
Writeln('           ',name[index],' to be varied from ',xmin:8:1,' to',xmax:9:1,
' in steps of ',xstep:8:2);  NormVideo;
Writeln('COLUMN VARIABLE : '); LowVideo;
Write('           ',name[jndex],' values :');
For i:=1 to jmax do Write(ppcol[i]:8:2);  NormVideo; Writeln;
Writeln('TARGET AC :               WITH AREAS (AX,AY,AZ) : '); LowVideo;
Write('           ',acname[iact],'                    ');
Writeln(tabl[iact,1]:7:1,tabl[iact,2]:7:1,tabl[iact,3]:7:1); NormVideo;
Writeln;
pp[index]:=xmin*conv[index]; pp[jndex]:=ppcol[1]*conv[jndex];
Writeln('CURRENT VALUES OF INPUT VARIABLES:');  Writeln;
Dispp(pp,ppe,conv,name,index,jndex);     {Display input variables}
Write('Type name of any variable to be changed  (CR if none)');
GetVar(st,i);
If (i>0) then
     begin
     Writeln('Currently ',name[i],'=',ppe[i]:10:2);
     Write('Type new value>>'); ReadR(nitems,buf); x:=buf[1];
     ppe[i]:=x; pp[i]:=x*conv[i];
     If (i=11) then iact:=Round(pp[11]); goto L1; end;
{ - - - - - - - - - - - - - - - - - - - - - - SET-UP COMPLETE.  RUN ANALYSIS - - }
Writeln('MENU :'); Writeln('       CR = RUN ANALYSIS');
Writeln('       1 = REDEFINE INPUT CONDITIONS');
Write('>>'); i:=-44; Readln(i);
If (i<>-44) then goto REDEF;
Write('Type note (CR if none)>>'); Readln(note);
{ - - - - - - - - WRITE HEADING INFO TO OUTPUT FILE OUTF - - - }
Assign(OutF,OutFNam); Rewrite(OutF);
Writeln(OutF,lm,'SEE1 VISUAL ACQUISITION ANALYSIS');
If (Length(note)>0) then Writeln(OutF,lm,'      | NOTE :',note);
Writeln(OutF,lm,'      | SEE1.PAS (Version ',version,')');
Writeln(OutF,lm,'      | InFNam= ',InFNam:17,'     VFNam= ',VFNam);
Writeln(OutF,lm,'      | ACFNam= ',ACFNam:17,'     OutFNam= ',OutFNam);
iact:=Round(pp[11]);
Writeln(OutF);
Writeln(OutF,lm,'INITIAL VALUES OF INPUT VARIABLES:');
Lstpp(pp,ppe,conv,name);
Writeln(OutF,lm,'ROW VARIABLE:');
Writeln(OutF,lm,lm,name[index],' is varied from',xmin:9:2,' to',
xmax:9:1,' in steps of',xstep:8:2);
Writeln(OutF,lm,'COLUMN VARIABLE :');
Write(OutF,lm,lm,name[jndex],' assumes values ');
For i:=1 to jmax do
   begin
   If (i=5) then
      begin Writeln(OutF); Write(OutF,lm,lm,lm,lm,lm,'     '); end;
```

```
      Write(OutF,ppcol[i]:8:2);
   end;
   Writeln(OutF);
Out_const(pp,ppe,conv,tabl,acname,index,jndex);
{ - - - - - - - - - - - - - - - - - - - - - START COLUMN ITERATION  - - - }
For icol:=1 to jmax do
begin
Writeln('   NOW COMPUTING WITH ',name[jndex],'=',ppcol[icol]:8:2);
pp[jndex]:=ppcol[icol]*conv[jndex];   pavg[icol]:=0.0;
If (long=1) then
   begin
   Writeln(OutF);
   Writeln(OutF,lm,'SEE1 ANALYSIS : PACQ VS. ',name[index],',    ',
   name[jndex],'=',ppcol[icol]:8:2);
   Writeln(OutF,lm,'      ',name[index],'   RDOT      AREA    THT     THS',
   '    RNG1    BEF1     PACQ');
   end;
{ - - - - - - - - - - - - - - - - - - - - START ROW ITERATION - - - - - - - }
ix:=0;  xval:=xmin-xstep;
While (xval<xmax) do
begin
ix:=ix+1;
xval:=xval+xstep;    ppe[index]:=xval;  pp[index]:=xval*conv[index];
Module1(pp[11],tabl,pp[18],pp[19],pp[20]);
Module2(pp[9],pp[10],pp[12],pp[17],pp[22],pp[23]);
Module3(pp[18],pp[19],pp[23],pp[21]);
Module4(pp[1],pp[2],pp[15],pp[16],pp[13],pp[14],pp[22],pp[26],pp[27],fov);
Module5(pp[4],pp[6],pp[17],pp[25]);
Module6(pp[3],pp[5],pp[17],pp[24]);
   IF (DEBUG>=2) THEN BEGIN WRITELN(OutF,' INPUT TO MODULE 7: AREA=',pp[21]:9:6,
   ' BEF0=',pp[26]:7:2,' BEF1=',pp[27]:7:2);
   WRITELN(OutF,'     DLIM=',pp[8]:7:2,' R=',pp[7]:7:2,' RDOT=',pp[17]:7:3); END;
Module7(pp[21],pp[26],pp[27],pp[8],pp[7],pp[17],pp[24],pp[25],pp[28]);
pavg[icol]:=pavg[icol]+pp[28];
Exunits(pp,ppe,conv);
If (long=1) then Writeln(OutF,lm,ppe[index]:9:1,ppe[17]:9:1,ppe[21]:9:1,
ppe[22]:7:1,ppe[23]:7:1,ppe[24]:7:2,ppe[27]:9:0,ppe[28]:9:4);
If (ix<32)  and (fov>0.0) then table[ix,icol]:=ppe[28]
else if (fov=0.0) then table[ix,icol]:=-1.0;
end;    {of index block}
{ - - - - - - - - - - - - - - - - - - - - - - - - END ROW ITERATION - - - - - }
pavg[icol]:=pavg[icol]/nrows;
end;   {of icol block}
{ - - - - - - - - - - - - - - - - - - - - - - - - END COLUMN ITERATION - - - -}
{ - - - - - - - - - - - - - - - - - - - - - - - - OUTPUT PACQ TABLE - - - - - }
Writeln(OutF,'PACQ TABLE : ');
Writeln(OutF);
Writeln(OutF,lm,'                         TABLE OF PACQ VALUES');
Writeln(OutF); Writeln(OutF,lm,'                                     ',name[jndex]);
```

```
Write(OutF,lm,'         ');
For i:=1 to jmax do Write(OutF,ppcol[i]:9:2); Writeln(OutF);
Write(OutF,lm,'  ',name[index],'   ');
For i:=1 to jmax do Write(OutF,'  ------ ');
Writeln(OutF);
ufov:=FALSE;
For i:=1 to ix do
     begin
     x:=xmin+(i-1)*xstep;  Write(OutF,lm,x:8:1);
     For j:=1 to jmax do
        If (table[i,j]>=0.0005) then Write(OutF,table[i,j]:9:4)
        else if (table[i,j]=-1.0) then
        begin Write(OutF,'    xfov ');  ufov:=TRUE; end
        else Write(OutF,'      *  ');
     Writeln(OutF);
     end;
pp[index]:=nomval[index]*conv[index];
pp[jndex]:=nomval[jndex]*conv[jndex];
Writeln(OutF); Write(OutF,lm,'    avg.');
For i:=1 to jmax do Write(OutF,pavg[i]:9:4);    Writeln(OutF);
If ufov then begin Writeln(OutF); Writeln(OutF,lm,lm,lm,lm,
'( Note : xfov = target not within field-of-view)'); end;
Writeln(OutF);
Writeln(OutF,lm,'*-*-*-* END OF ANALYSIS  *-*-*-*');
Close(OutF);
Writeln('OUTPUT STORED IN DISK FILE [',OutFNam,']');
{ - - - - - - - - - - - - - - - - - - - - - - - -SELECT MODE FOR OUTPUT - - - }
Writeln('MENU:'); Writeln('     1 = print results');
Writeln('     2 = display results'); Writeln('     CR = continue');
Write('>>'); menu:=0; Readln(menu);
If (menu>0) then
   begin
   ClrScr;
   If (menu=1) then AuxOutPtr:=LstOutPtr else AuxOutPtr:=ConOutPtr;
   Assign(OutF,OutFNam); Reset(OutF);
   While not Eof(OutF) do
      begin
      Readln(OutF,s);
      {If (Length(s)>85) then WRITELN('         LENGTH(S)=',Length(s):4);}
      ss:=Copy(s,1,10);
      If (ss='PACQ TABLE') and (menu=2) then
         begin Write('CR to view PACQ table>>'); Readln; ClrScr; end
         else
            begin
            If (menu=2) and (Copy(s,1,5)='     ') then s:=Copy(s,6,250);
            Writeln(Aux,s);
            end;
      end;
   Close(OutF);
   end;
```

```
Write('CR to continue>>');  Readln;
{ - - - - - - - - - - - - - - - - - - - - - -SELECT TERMINATION OPTIONS - - - }
Writeln('MENU :');
Writeln('        1 = NEW ANALYSIS (KEEP PREVIOUS MODS)');
Writeln('        2 = NEW ANALYSIS (REINITIALIZE ALL INPUTS)');
Writeln('       CR = QUIT');
Write('>>');  menu:=0; Readln(menu);
If (menu=1) then goto L1 else if (menu=2) then goto START;
Close(OutF);
Writeln(' OUTPUT SAVED IN ',OutFNam);
end.
```

C.2 Listing for SEE2

This section provides a listing of the Pascal code for SEE2. Each procedure is shown seprately in alphabetical order of name. Those procedures that are identical to those used in SEE1 are not included (they are provided in section C.2).

```
PROCEDURE Dispp2(Var pp,pp3,pp4,conv: ppty; name: namety;
xndex,yndex: integer);
     {DISPLAY INPUT VARIABLES TO CONSOLE}
VAR stx: string[255];
begin
Writeln('            - REFERENCE PAIR -    DUMMY    SUBJECT');
Writeln('               AC1        AC2        AC3        AC4');
For i:=1 to 16 do
   begin
   stx:=name[i]+'          ----       ----       ----       ----     ';
   If (i=xndex) then Writeln(stx,' ROW VARIABLE')
   else if (i=yndex) then Writeln(stx,' FLOATING VARIABLE')
   else Writeln(StDisp(i,pp,pp3,pp4,conv,name));
   end;
Writeln;
end;      {of PROCEDURE Dispp2  Revised 7/30a/87}


PROCEDURE Make_Consistent(Var pp,pp3,pp4: ppty);
{- - - - - MAKE V1,V2,XANG,AND R CONSISTENT BETWEEN PAIRS}
begin
pp3[9]:=pp4[10]; pp3[10]:=pp4[9];  pp3[7]:=pp4[7];  pp[7]:=pp4[7];
pp3[12]:=pp4[12];
end;
```

```
FUNCTION StDisp(index: integer; Var pp,pp3,pp4,conv: ppty;
Var name: namety): str255;
Var vout: array[1..4] of real;  i,format: integer; st,sx: string[255];
{ RETURN PP VARIABLE PRINT-OUT LINE FOR 4 AC }
begin
vout[1]:=pp[index]/conv[index];
vout[2]:=pp[index]/conv[index];
vout[3]:=pp3[index]/conv[index];
vout[4]:=pp4[index]/conv[index];
st:=name[index]+'     ';
format:=1;
Case index of
   1..2  : format:=0;      5..6  : format:=2;
   8     : format:=2;      11    : format:=0;
   15..20 : format:=0;
   end;
For i:=1 to 4 do
   begin
   If (format=1) then Str(vout[i]:10:1,sx)
   else If (format=0) then Str(vout[i]:10:0,sx)
   else Str(vout[i]:10:2,sx);
   st:=st+sx;
   end;
StDisp:=st;
end; {of Function StDisp}


PROCEDURE Outpp(Var pp,pp3,pp4,ppe,conv: ppty; Var name: namety;
xndex,yndex: integer);
{Print input variables to OutF}
Var stx : string[255];
begin
Writeln(OutF,'             - REFERENCE PAIR -    DUMMY    SUBJECT');
Writeln(OutF,'               AC1        AC2        AC3        AC4');
For i:=1 to 16 do
   begin
   stx:=name[i]+'      ----      ----      ----      ----   ';
   If (i=xndex) then Writeln(OutF,stx,'  ROW VARIABLE')
   else if (i=yndex) then Writeln(OutF,stx,'  FLOATING VARIABLE')
   else Writeln(OutF,StDisp(i,pp,pp3,pp4,conv,name));
   end;
Writeln(OutF);
end;        {of PROCEDURE Outpp}
Program SEE2;   {Runs from default disk}
Const
   version='B03'; sqnmi=2.70872E-08;  nmisec=2.77777E-04;
   default=0.44444;    debug=0;
```

```
Label  ENDITT,L1,L2,REDEF,START;
Type
     str255 = string[255];
     namety = array[1..32] of string[4];
     ppty = array[1..32] of real;
     rbuff = array[1..32] of real;
     strng4 = string[4];  strng255 = string[255];
     tabty = array[1..16,1..4] of real;
Var
     TFile,OutF,InF : text;
     area,ax,ay,az,bef0,bef1,bet0,bet1,dfdy,dlim,dy,d1,d2 : real;
     fnew,fold,fovl,fovr,fstop,pacq,pacqref,pl,pr : real;
     r,rdot,rng1,rng2,dystop,t1,t2,th1,th2,v1,v2 : real;
     x,xang,xmin,xmax,xsmax,xstep,xval,y,ye,yfirst,yll : real;
     ynew,yold,ystep,yul,y0 : real;
     i,ii,iac,iact,index,iLST,ix,j,jmax : integer;
     nacty,nitems,nitt,nstop,xndex,yndex : integer;
     conv,nomval,pp,pp3,pp4,ppe : ppty;       buf : rbuff;
     table : array[1..32,1..8] of real;
     name : namety;    st : string[4];
     ACFNam,InFNam,OutFNam,SetFNam,VFNam : string[12];
     acname : array[1..16] of string[12];   note,s,title : strng255;
     acid : array[1..16] of integer;
     pacqac,pcol : array[1..8] of real;
     tabl : tabty;
{$I SEEMODS.B03}
PROCEDURE Make_Consistent(Var pp,pp3,pp4 : ppty);
{- - - - - MAKE V1,V2,XANG,AND R CONSISTENT BETWEEN PAIRS}
begin
pp3[9]:=pp4[10]; pp3[10]:=pp4[9];  pp3[7]:=pp4[7];  pp[7]:=pp4[7];
pp3[12]:=pp4[12];
end;
FUNCTION StDisp(index : integer; Var pp,pp3,pp4,conv : ppty;
Var name : namety) : str255;
Var vout : array[1..4] of real;  i,format : integer; st,sx : string[255];
{ RETURN PP VARIABLE PRINT-OUT LINE FOR 4 AC }
begin
vout[1]:=pp[index]/conv[index];
vout[2]:=pp[index]/conv[index];
vout[3]:=pp3[index]/conv[index];
vout[4]:=pp4[index]/conv[index];
st:=name[index]+'      ';
format:=1;
Case index of
   1..2  : format:=0;       5..6  : format:=2;
   8     : format:=2;       11    : format:=0;
   15..20 : format:=0;
   end;
```

```
For i:=1 to 4 do
   begin
   If (format=1) then Str(vout[i]:10:1,sx)
   else If (format=0) then Str(vout[i]:10:0,sx)
   else Str(vout[i]:10:2,sx);
   st:=st+sx;
   end;
StDisp:=st;
end; { of Function StDisp}
PROCEDURE Dispp2(Var pp,pp3,pp4,conv : ppty; name : namety;
xndex,yndex : integer);
     {DISPLAY SEE2 INPUT VARIABLES TO CONSOLE }
VAR stx : string[255];
begin
Writeln('              - REFERENCE PAIR -    DUMMY    SUBJECT');
Writeln('                 AC1        AC2       AC3       AC4');
For i:=1 to 16 do
   begin
   stx:=name[i]+'        ----      ----      ----      ----    ';
   If (i=xndex) then Writeln(stx,'  ROW VARIABLE')
   else if (i=yndex) then Writeln(stx,'  FLOATING VARIABLE')
   else Writeln(StDisp(i,pp,pp3,pp4,conv,name));
   end;
Writeln;
end;      {of PROCEDURE Dispp2  Revised 7/30a/87}
PROCEDURE Outpp(Var pp,pp3,pp4,ppe,conv : ppty; Var name : namety;
xndex,yndex: integer);
{PRINT INPUT VARIABLES TO OUTPUT FILE, OUTF}
Var stx : string[255];
begin
Writeln(OutF,'            - REFERENCE PAIR -    DUMMY    SUBJECT');
Writeln(OutF,'               AC1        AC2       AC3       AC4');
For i:=1 to 16 do
   begin
   stx:=name[i]+'        ----      ----      ----      ----    ';
   If (i=xndex) then Writeln(OutF,stx,'  ROW VARIABLE')
   else if (i=yndex) then Writeln(OutF,stx,'  FLOATING VARIABLE')
   else Writeln(OutF,StDisp(i,pp,pp3,pp4,conv,name));
   end;
Writeln(OutF);
end;       {of PROCEDURE Outpp}
begin  {- - - -  - - - -  - - - - - - - - - - - - - - - - SEE2 MAIN PROGRAM - - }
Writeln('*-*-*-* SEE2 VISUAL ACQUISITION ANALYSIS  (Version ',version,') *-*-*-*');
{- - - - READ DATA FROM INITIALIZATION FILE - - - - }
InFNam:='SEE2SET.B03';     Writeln('      | INIT FILE HEADING :',InFNam);
Assign(InF,InFNam);  Reset(InF);
Readln(InF,s);       Writeln('      | INIT FILE TITLE : [',s,']');
Readln(InF,SetFNam); Readln(InF,VFNam); Readln(InF,ACFNam);
Readln(InF,OutFNam);  Write('      | SetFNam=',SetFNam);
Writeln('     VFNam =',VFNam);     Write('      | ACFNam=',ACFNam);
Writeln('     OutFNam=',OutFNam);
```

```
Readln(InF,acid[1],acid[2],acid[3],acid[4],acid[5],acid[6]);
s:=''; For i:=1 to 6 do If (acid[i]>0) then s:=s+Chr(acid[i]);
Write(Lst,s);
Assign(OutF,OutFNam); Rewrite(OutF);
{ - - - - - - - - - - - - - - READ IN VARIABLE NAMES AND NOMINAL VALUES - - - -}
Assign(TFile,VFNam);  Reset(TFile);
Readln(TFile);    {Skip label line}
For i:=1 to 28 do
Readln(TFile,name[i],nomval[i],conv[i]);
Close(TFile);
{- - - - -  - - - - - - - - - - - - - - READ IN DATA FROM SET-UP FILE - - - - }
Writeln('Type name of file containing initialization data.');
Write('(CR for ',SetFNam,') >>');
 Readln(s);   If (Length(s)>0) then SetFNam:=s;
Assign(TFile,SetFNam); Reset(TFile);
yfirst:=default;  yndex:=9;
Readln(TFile,s); Writeln(s);
Readln(TFile,title); Writeln(title);
Readln(TFile,st,xndex,xmin,xmax,xstep);
Readln(TFile,yll,yul,dystop);
Readln(TFile);
For i:=1 to 16 do
   begin
   Readln(TFile,st,x,pp[i],pp3[i],pp4[i]);
   end;
For i:=1 to 16 do
   begin
   pp[i]:=pp[i]*conv[i];  pp3[i]:=pp3[i]*conv[i]; pp4[i]:=pp4[i]*conv[i];
   end;
Close(TFile);
{- - - - READ TABLE OF AIRCRAFT CHARACTERISTICS - - - - }
Assign(TFile,ACFNam); Reset(TFile);
nacty:=0;
While  not Eof(TFile) do
   begin
   nacty:=nacty+1;
   Readln(TFile,acid[nacty],acname[nacty],tabl[nacty,1],tabl[nacty,2],
   tabl[nacty,3]);
   end;
Close(TFile);
Make_Consistent(pp,pp3,pp4);
{ - - - - - - - - - - - PRINT-OUT SET-UP DATA - - - - - - }
Writeln('SET-UP DATA FROM FILE ',SetFNam,':');
Dispp2(pp,pp3,pp4,conv,name,xndex,yndex);
Writeln('FLOATING VARIABLE: ',name[yndex],' for AC4 will vary to force equivalency');
Writeln('ROW VARIABLE: ',name[xndex],' stepped from',xmin:8:1,' to',xmax:9:1);
Writeln('      in steps of',xstep:9:2);
REDEF :
{- - - - - - - - - - - - - - - - - - - ALTER EXISTING PP VALUES - - - - - - - - - }
```

```
L1 :
Write('Type name of any variable to be changed  (CR if none)');
GetVar(st,i);
If (i=xndex) or (i=yndex) then
   Writeln(Chr(7),' WARNING : THIS IS A PROGRAM-CONTROLLED VARIABLE!');
If (i>0) then
      begin
      Write('Type AC to change (1-4):>>'); Readln(iac);
      Write('Type new value>>'); ReadR(nitems,buf); x:=buf[1];
      If (iac<3) then  pp[i]:=x*conv[i]
         else If (iac=3) then pp3[i]:=x*conv[i]
         else If (iac=4) then pp4[i]:=x*conv[i];
      Make_Consistent(pp,pp3,pp4);
      Writeln('REVISED SET-UP VARIABLES :');
      Dispp2(pp,pp3,pp4,conv,name,xndex,yndex);
      goto L1;
      end;
{ - - - - - - - - - - - - - - - - - WRITE HEADER INFO TO OUTPUT FILE, OUTF - - - }
Write('Type note (CR if none)>>'); Readln(note);
Writeln(OutF,'*-*-*-* SEE2 VISUAL ACQUISITION ANALYSIS *-*-*-*');
If (Length(note)>0) then Writeln(OutF,'NOTE:',note);
Writeln(OutF,'       | SEE2 VERSION: ',version);
Writeln(OutF,'       | INPUT DATA SET NAME: ',InFNam);
Writeln(OutF,'       | INPUT DATA SET TITLE: ',title);
Writeln(OutF,'       | SetFNam=',SetFNam,'     VFNam =',VFNam);
Writeln(OutF,'       | ACFNam=',ACFNam,'     OutFNam=',OutFNam);
Writeln(OutF);
Writeln(OutF,'INITIAL VALUES OF INPUT VARIABLES:');
Outpp(pp,pp3,pp4,ppe,conv,name,xndex,yndex);
Writeln(OutF);
Writeln(OutF,'ROW VARIABLE: ',name[xndex],' varies from',xmin:9:2,
' to',xmax:9:1,' in steps of',xstep:9:2);
WRITELN(OutF,'FLOATING VARIABLE: ',name[yndex],' (between limits of' ,
yll:7:2,' and ',yul:9:2,')');
Writeln(OutF);
Writeln(OutF,'AIRCRAFT TYPES:');
Writeln(OutF,'             ACID  TYPE            AX      AY      AZ');
For i:=1 to 4 do
   begin
   If (i<3) then iact:=Round(pp[11]) else If (i=3) then
   iact:=Round(pp4[11]) else iact:=Round(pp3[11]);
   Writeln(OutF,'     AC',i:1,acid[iact]:6,'  ',acname[iact],tabl[iact,1]:8:1,
   tabl[iact,2]:8:1,tabl[iact,3]:8:1);
   end;
   Writeln(OutF);
Writeln(OutF,'  - - - - - -  OUTPUT TABLE  - - - - - - - -');
{- - - - - - - - - - - - - - - - - - - - - START COMPUTATIONAL BLOCK- - - - - - }
ix:=0;  xval:=xmin-xstep;  y0:=pp[yndex];
While (xval<xmax) do
begin
ix:=ix+1;
```

```
xval:=xval+xstep;    pp4[xndex]:=xval*conv[xndex];
Writeln('   >Computing for ',name[xndex],'=',xval:10:1);
Make_Consistent(pp,pp3,pp4);
{-------- FIND REFERENCE PACQ ----------}
pacqac[1]:=Fpacq(pp,tabl);  pacqac[2]:=pacqac[1];
pacqref:=1.0-(1.0-pacqac[1])*(1.0-pacqac[2]);
{- -  ITERATE PP4[YNDEX] TO OBTAIN EQUIVALENCY FOR SECOND  PAIR OF AC--------}
nitt:=0;  nstop:=10;
y:=pp4[yndex];  ye:=y/conv[yndex];  dy:=10*dystop*conv[yndex];
While (nitt<nstop) and (Abs(dy)>dystop*conv[yndex]) do
   begin    {BEGIN NITT LOOP }
   nitt:=nitt+1;
   IF (KEYPRESSED) THEN BEGIN WRITE('CR TO CONTINUE'); READLN(I); END;
   IF (debug>=2) THEN BEGIN
   WRITELN(OutF);
   WRITELN(OutF,' - - - - BEGIN ITERATION NO. ',NITT:3,' - - - - - - - - ');
   WRITELN(OutF,'      VALUE OF ',name[yndex],' IS NOW',y:10:4);  END;
   fold:=fnew;  yold:=ynew;  ynew:=y;
   pacqac[3]:=Fpacq(pp3,tabl);  pacqac[4]:=Fpacq(pp4,tabl);
   IF (debug>2) then WRITELN(OutF,'    FOR ',name[yndex],' = ',ye:7:2,' kt',
   '  pacqac[4]=',pacqac[4]:9:5);
   fnew:=1.0-(1.0-pacqac[3])*(1.0-pacqac[4]);
   If (nitt>1) then
      begin
      If (ynew=yold) then Writeln(Chr(7),'!!CONVERGENCE PROBLEM : YNEW=YOLD=',
           YNEW:10:4,' AT Y=',Y:12:4)

      else If (fnew=fold) then
           Writeln(OutF,Chr(7),'!! CONVERGENCE PROBLEM: fnew=fold=',fnew:9:5)
      else
         begin
         dfdy:=(fnew-fold)/(ynew-yold);
         dy:=(pacqref-fnew)/dfdy
         end;
      end;
   y:=y+dy;   ye:=y/conv[yndex];
   If (ye<yll) then ye:=yll;   If (ye>yul) then ye:=yul;
   y:=ye*conv[yndex];  pp4[yndex]:=y;
   Make_Consistent(pp,pp3,pp4);
   If (debug>2) then
      begin
      WRITE(OutF,' NITT        YNEW        DY       FNEW       FOLD      DFDY');
      WRITELN(OutF,'    PACQAC[3]    PACQAC[4]');
      WRITE(OutF,NITT:4,YE:10:3,'kt   ',DY/conv[yndex]:8:3,'kt',FNEW:10:4,' ',
      FOLD:8:4,' ',dfdy:9:4);
      WRITELN(OutF,pacqac[3]:10:5,pacqac[4]:10:5);
      end;
   end; {of nitt loop}
```

```
ENDITT:
If (nitt=nstop) or (y=yll) or (y=yul) then
   begin
   Writeln(OutF);
   Writeln(OutF,'EQUIVALENCY SEARCH FAILED FOR X =',xval:10:4,' ( nitt,y=',
   nitt:4,y:12:4,')');
   end;
{ - - - - - - - - - - - - - - - - - - - - - - END COMPUTATIONAL BLOCK - - - - - }
L2:
If (xval=xmin) or (debug>0) then Writeln(OutF,'        ',name[xndex],'          ',
name[yndex],'   pacqref   pacq3    pacq4 ');
Writeln(OutF,'<<',xval/conv[xndex]:11:3,y/conv[yndex]:10:2,
pacqref:9:4,pacqac[3]:9:4,pacqac[4]:9:4,' >>');
end;    {of x index block}
{pp[yndex]:=nomval[yndex]*conv[yndex];  }
Writeln(OutF);
Writeln(OutF,'*-*-*-* END OF SEE2 ANALYSIS  *-*-*-*');
Close(OutF);
Writeln('Analysis Complete - results stored in ',OutFNam);
Writeln('MENU:'); Writeln('     1 = print results');
Writeln('     2 = display results'); Writeln('     CR = continue');
i:=0; Readln(i);
If (i>0) then
   begin
   If (i=1) then AuxOutPtr:=LstOutPtr else AuxOutPtr:=ConOutPtr;
   Assign(OutF,OutFNam); Reset(OutF);
   ii:=0;
   While (not Eof(OutF)) do
      begin
      Readln(OutF,s); Writeln(Aux,s); ii:=ii+1;
      If (ii>=12) then begin Delay(2000); ii:=0; end;
      end;
   Close(OutF);
   end;
Write('CR to continue');  Readln(i);
i:=0; Writeln('MENU :');
Writeln('        1 = NEW ANALYSIS (MODIFY PARAMETERS)');
Writeln('       CR = QUIT');
Write('>>');  Readln(i);
If (i=1) then goto REDEF;
end.
```