REPORT NO. DOT-TSC-OST-71-12

AN AIRPORT AIRSIDE SYSTEM MODEL

IRWIN ENGLANDER
DATA TECHNOLOGY BRANCH
COMPUTER TECHNOLOGY DIVISION
TRANSPORTATION SYSTEMS CENTER
55 BROADWAY
CAMBRIDGE,MA 0 2142

JUNE 1971
TECHNICAL REPORT

Prepared for

OFFICE OF THE SECRETARY
U.S. DEPARTMENT OF TRANSPORTATION
WASHINGTON, D.C. 20590

1. Report No. DOT-TSC-OST-71-12	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle An Airport Airside System Model		5. Report Date June 1971
		6. Performing Organization Code TCD
7. Author(s) Irwin Englander		8. Performing Organization Report No.
9. Performing Organization Name and Address Transportation Systems Center 55 Broadway Cambridge, Ma. 02142		10. Work Unit No. R-1106
		11. Contract or Grant No. N/A
12. Sponsoring Agency Name and Address Office of the Secretary U.S. Department of Transportation Washington, D.C. 20590		13. Type of Report and Period Covered Technical Report
		14. Sponsoring Agency Code OS-04
15. Supplementary Notes		

16. Abstract

This model of an airport airside system simulates aircraft operations and controller functions in the terminal area, both in the air and on the ground. The model encompasses all operations between the terminal gate and the point of handoff between the enroute controller and the terminal controller.

17. Key Words	18. Distribution Statement Unclassified

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

TABLE OF CONTENTS

	Page
ACKNOWLEDGEMENTS	iv
INTRODUCTION	1
Operation of the Model	1
Applications of the Model	2
Limitations of the Model	3
Organization of This Report	3
INPUT	6
Airport Design Module	8
Control Rules Module	8
Arrival and Departure Schedule Module	8
OUTPUT	9
Statistics Module	9
Graphics Output	9
RADAR CONTROLLER	10
Arrival Aircraft Module	10
Departure Aircraft Module	12
LOCAL CONTROLLER	13
Ground Controller Module	13
Arrival Local Controller Module	15
Departure Local Controller Module	15
AIRCRAFT ARRIVAL	16
Flight Origin Module	16
Area Control Module	16
Approach Pattern Module	16
Taxiway and Flight Termination Module	18
AIRCRAFT DEPARTURE	19
Taxiway and Flight Start Module	19
Departure Aircraft Control Module	19
Departure Pattern Module	19
REFERENCES	22
APPENDIX A	A-0

LIST OF ILLUSTRATIONS

Page

1. Airport Airside Model 5
2. Input 7
3. Radar Controller 11
4. Local Controller 14
5. Aircraft Arrival 17
6. Aircraft Departure 20

ACKNOWLEDGMENTS

The author is indebted to the many persons who have assisted him in learning about the terminal air traffic control system. Among these are L. Higgins of TSC/SA, G. Bates of RD 700/SRDS/F.A.A., R. L. Bierman of RD 620/SRDS/F.A.A. and G. Halverson NA 520/NAFEC. In addition, there have been comments by other members of the FAA and TSC with whom I have discussed the work, but whose contributions are not readily identifiable.

I also want to thank A. Zellweger and A. Priver of TCD/TSC who programmed the data link communications and computer graphics.

Finally, I would like to thank E. M. Darling, Jr. TCD/TSC who patiently edited this report and converted the rough ideas into a readable report.

INTRODUCTION

An Airport Airside System model, simulating the airport control system from the terminal gate to the point of handoff to the enroute controller, has been developed for use in analyzing system performance in response to a variety of operating rules, control procedures, runway and taxiway configurations, and distributions of aircraft types. The model consists of an Input Package for defining the data and parameters, an Output Package to compute performance measures and to handle graphics displays, Arrival and Departure Packages to move planes through the system, and Local and Radar Controller Simulation Packages (both having arrival and departure portions) to control the movements of aircraft.

The model exists in two forms: batch and interactive. Both allow the user to analyze a problem quickly and easily. Experience with the model will provide insight into both its strengths and weaknesses and, more importantly, point the way to future upgrading of the model in the direction of a closer approximation to the real world. The interactive version of the model utilizes a computer graphics system which provides the user with a natural and easy means for setting up the problem, watching the system in action, changing parameters, and gathering performance statistics.

OPERATION OF THE MODEL

The model permits the user to enter any desired airport surface configuration, consisting of a set of runways and taxiways. All altitude levels can be used by aircraft flying in the system, and standard vertical and horizontal separation criteria for aircraft are incorporated.

For simplicity, this first model is restricted to aircraft operating under Instrument Flight Rules (IFR). Furthermore, all flights in the system conform to schedules (which are defined by the user for each type of aircraft). Multiple flights may be scheduled for departure at the same time on one runway. These flights are then sequenced according to the characteristics of the runway and aircraft being used, taking into consideration the number of incoming flights and the rules governing aircraft separation in the terminal area. Any pattern may be used and the only restriction is that handoff points conform to Standard Instrument Departure (SID) routes.

Arriving aircraft enter the airport area through a simulated handoff to a radar controller and are then stacked for later processing if necessary or, if traffic load permits, are immediately directed into a controlled pattern for landing and taxiing along prescribed routes to specified gates. Departing aircraft leave gates under ground control, take off under local control, and proceed under radar control up to the point of simulated handoff to enroute control. Flights are continuously updated while in progress and search logic in the Radar Controller simulation detects projected, or actual, violations of the separation rules. Automatic conflict resolution is an option in the model.

Standard air traffic controller functions exist, including communications between aircraft and controllers, as well as, provisions for altering flight paths because of potential conflicts. A list of potential conflicts (a potential conflict is a situation where two aircraft are less than five miles apart and closing) requiring controller attention is generated, and any potential conflict which would be apparent to the controller can be resolved automatically, if desired.

Measures of system performance are calculated while the model is in operation. These include controller workload, arrival rate, departure rate, queue size, etc. The system in action will be portrayed on interactive graphics displays in a form which is both realistic and readily understandable. In particular, areas of congestion or potential hazard can be immediately seen. Moreover, the model will be designed to be useable by people without computer programming experience.

APPLICATIONS OF THE MODEL

The model can be used to analyze the effect of various airport runway configurations, traffic patterns, aircraft mixes, etc., on airport capacity. For instance, the batch version of the model has already been used to evaluate the effect of intermixing STOL and conventional (CTOL) aircraft in the traffic pattern of a typical airport. Realistic operational parameters provided by the Aircraft Division, SRDS, were used in this study, including STOL and CTOL aircraft characteristics, runway geometry for a particular airport, aircraft glide slope angles, and offset approach configurations. The performance measures used were arrival rate, departure rate, and number of conflicts.

Another major application is the analysis of acceptance rate, wherein airport arrival rate can be adjusted in accordance with controller work load. The model user can enter any acceptance rate algorithm he desires and evaluate its effect on airport operations (i.e. arrivals and departures).

A conflict analysis can be performed in which various measures of conflict potential are tested. Several thresholds (which depend on aircraft spacing and relative velocity) are possible and are available as options in the model. A vehicle access control system (VACS) simulator is being incorporated in the model for use in studying airport surface traffic control problems.

The above is not an exhaustive list of applications for the initial model. It is, rather, a representative list of typical problems which could be solved. We expect that other similar applications will suggest themselves and that these can be handled either directly, or by making minor modifications to the present model.

LIMITATIONS OF THE MODEL

The current model does not include all features of a real world Airport Airside System. The missing features are not omitted because of any inherent difficulty in programming them. Rather, their omission serves to reduce the complexity of the model. It would be a simple matter to later add any desired feature.

Specifically, VFR flights and emergencies are excluded from the system since they neither comprise a significant proportion of the scheduled air carrier traffic, nor does the ATC system exert as much influence on them as on IFR flights.

Also, while provisions exist for simulating hardware in detail, these have not been implemented in the present model. Furthermore, there is no intent to incorporate these features until a specific problem requires them.

ORGANIZATION OF THIS REPORT

This report is divided into three sections, one for each major function of the model, namely: Input/Output, Controller Actions, and Aircraft Movement and Communications. In addition, a complete listing of the simulation programs in FORTRAN IV is included in the Appendix. Figure 1 is a flow diagram of the model in which the portions constituting each of its three major functions are surrounded by dashed lines; the modules within each portion are shown as boxes labeled with a numeric code; and the flow of information is indicated by lines and arrows. Note that the Input and Output portions are replicated in the Arrival and Departure portions of the model. The gross organization of the model is as follows:

Major Functions	Portions
Input/Output	Input Output
Controller Actions	Radar Controller Local Controller
Aircraft Movement and Communications	Aircraft Arrival Aircraft Departure

Each section begins with a block diagram showing the inputs and outputs of all the modules contained in the portion of the model being described.

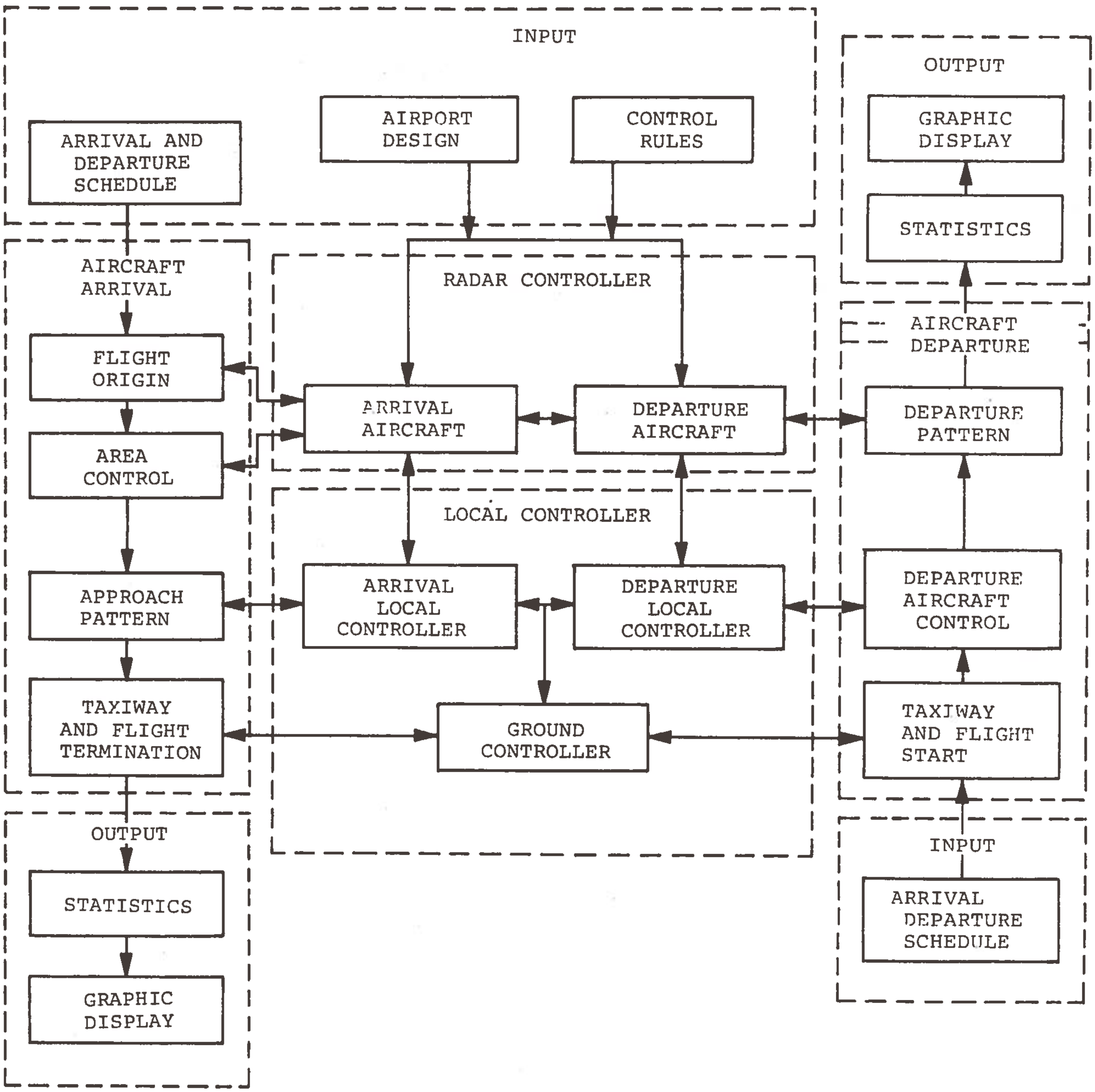

Figure 1. Airport Airside Model

INPUT

The input portion of the model (Figure 2) contains all the parameters which specify the airport design, control rules, arrival schedules, and departure schedules. For ease in actual operation the parameters are divided into two groups: those which are infrequently changed and hence are built into the program; those which are frequently changed and thus are entered as input each time a new problem is simulated. However, any variable can be readily transferred from one group to the other.

Among the built-in parameters are SID, STARS, holding patterns, ATC rules and controller workload maximums. Among the input parameters are runway and taxiway location, handoff points, and aircraft characteristics, (i.e. turn rate, climb rate, acceleration, decent rate, and wake separation).

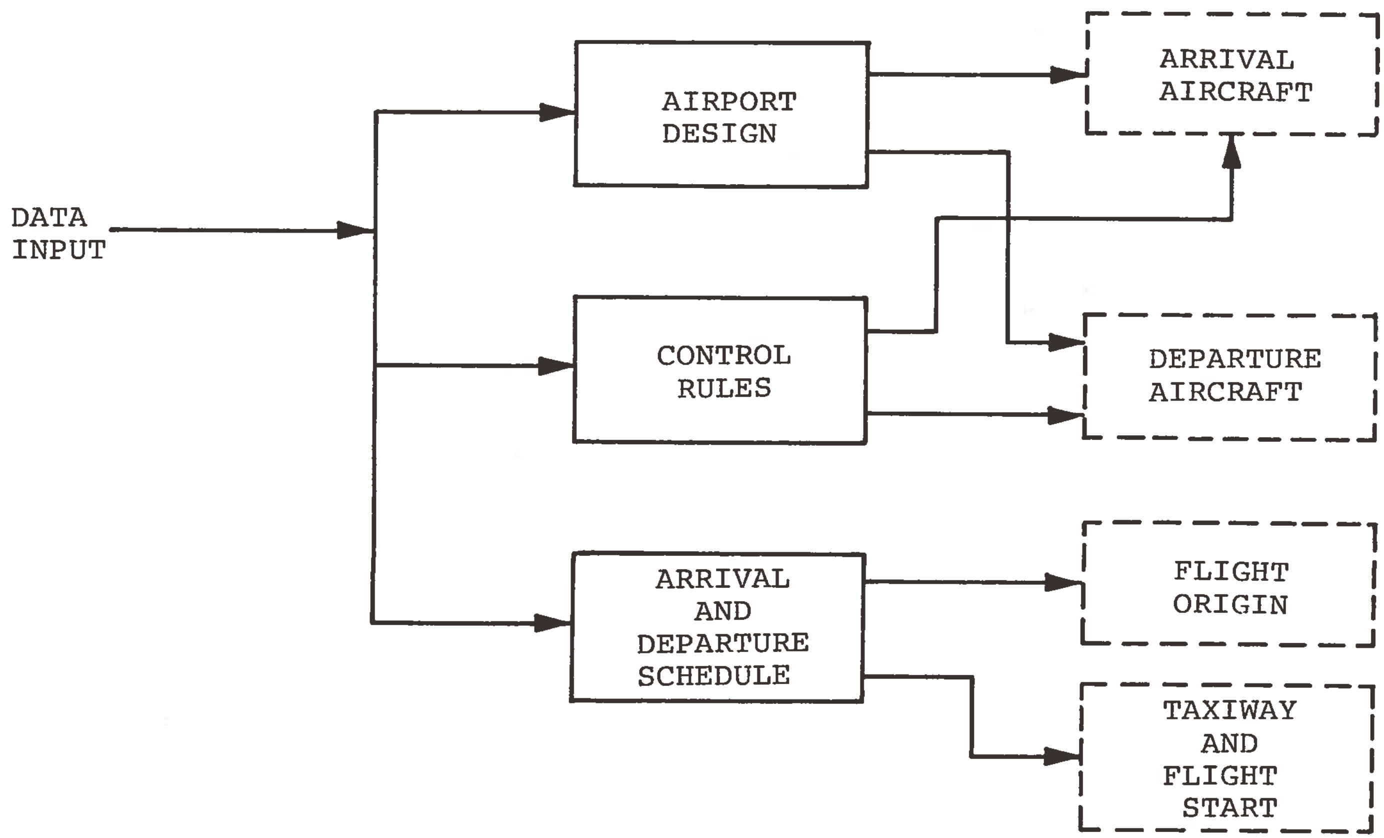

Figure 2. Input

AIRPORT DESIGN MODULE

The airport design consists of the boundaries where handoffs are made between the enroute controller and the airport controllers; the runway locations and lengths; the traffic distribution for each type of aircraft (i.e. numbers of conventional aircraft, STOL aircraft, or VSTOL aircraft); and the positions of outer markers.

CONTROL RULES MODULE

The Control Rules Module sets both the altitude at the outer marker and the glide path angle for each runway. Within each area a vector direction is defined for all points in the area below the holding pattern; the number of aircraft approaching a runway is set; a holding altitude is established and the holding pattern is protected. Maximum controller workload is an input which controls acceptance rate.

The spacing criteria for aircraft is three miles and one thousand feet altitude. A conflict is defined as a violation of these spacing criteria. (This is in conformance with F.A.A. rules). In addition, provisions are made for inputting standard instrument departures (SID) together with the requirement that the handoff of a departing aircraft can only occur when that aircraft is near or on a SID and outside the airport area.

ARRIVAL AND DEPARTURE SCHEDULE MODULE

This arrival part of this module allows the user to input an arrival schedule for aircraft by type (CTOL, STOL, VSTOL). In the current model, arrivals of each aircraft type are equally spaced with no randomness.

The departure schedule delivers aircraft to the runway line up in a similar manner. However, the actual takeoff depends upon the air traffic control situation.

OUTPUT

The output portion of the program has not been fully specified since each user may require different information depending upon his purpose. The program affords the user the flexibility of specifying what statistics will be collected and to select the output format (i.e. either a printout, a C.R.T. display, or both). The current user choices are specified below.

STATISTICS MODULE

During the operation of the program a variety of statistics are calculated. For each operating runway a count of operations is made. (The counter is incremented for arrivals when the aircraft first occupies the runway, whereas for departures the counter is incremented when the runway is cleared.)

In order to evaluate controller workload, the following statistics are gathered. First, a count of vectors issued by the simulated controller is made. (A vector is assumed to be issued whenever the logic of the program requires a turn or an altitude change.) A second count is conflict minutes. (For each interval of simulation time a count of pairs of aircraft in conflict is made. This count is multiplied by the simulation interval, and the product is added to the counter.) A third count is the conflict minutes per mile. (At each interval of simulation time the simulation interval is multiplied by the sum of the inverse of the separations, and the product is added to the counter.) This count has the advantage over a straight time count in that it weighs conflict according to their seriousness as shown by separation.

A count of the line up for each runway at each time interval is made to give a measure of departure delay. A count of stack size at each time interval is made to measure arrival delay.

GRAPHICS OUTPUT

The graphics output consists of two airport displays: one planer and one vertical. Both displays show the runways as well as all aircraft currently in the system. (At the user's option, an identification code can be displayed next to each aircraft.) Also, a table of statistics can be displayed on the planer view. To assist in coordination of the two views any group of aircraft can be made to blink, or become invisible, on both displays.

RADAR CONTROLLER

The Radar Controller portion of the model (Figure 3) is divided into two modules: the Arrival and Departure Aircraft Modules. The first controls arriving aircraft from the airport boundary until the aircraft is assigned to a runway and is handed off to the Arrival Local Controller. The second handles departing aircraft from the point where they are received by handoff from the Departure Local Controller up to the point where they are handed off to an enroute controller at the airport boundary. Both modules use a common conflict resolution routine since conflicts checking is done for all aircraft.

ARRIVAL AIRCRAFT MODULE

As each aircraft arrives, it is assigned to a runway which handles that type of aircraft and which has the smallest queue. If the assigned number for a given runway exceeds a specified limit (i.e., a variable, but presently set at five aircraft), the aircraft is placed in a holding pattern). (Each such pattern has a minimum holding altitude of 5000 feet and is allocated four fixes. For convenience, the model handles all aircraft in each holding pattern.) The number of aircraft in the holding pattern determines the altitude of the next arrival. Four oval holding patterns, each twenty miles long, are distributed around the airport. Aircraft enter these patterns on a first come first serve basis. The only exception is STOL aircraft which may have a separate holding pattern when the STOL runway is in operation.

If there is no holding required (or if an aircraft is being removed from the holding pattern for landing), the aircraft is assigned a vector bringing it to an initial approach altitude (currently 2200 feet for the first arrival runway and 3200 feet for a parallel runway). The aircraft is then vectored to a prescribed point for handoff to the Arrival Local Controller. The primary path is a function of the aircraft turn rate and descent rate.

A list of all pairs of aircraft and their altitudes is generated. Those pairs within three miles separation and one thousand feet altitude are classified as conflicts requiring priority attention. (Other separation distances can be designated as potential conflicts if desired.) The user has the option of moving aircraft on SID or STARS and merely recording conflicts, or he may choose to utilize an automatic conflict resolution program (available in the current model) which acts

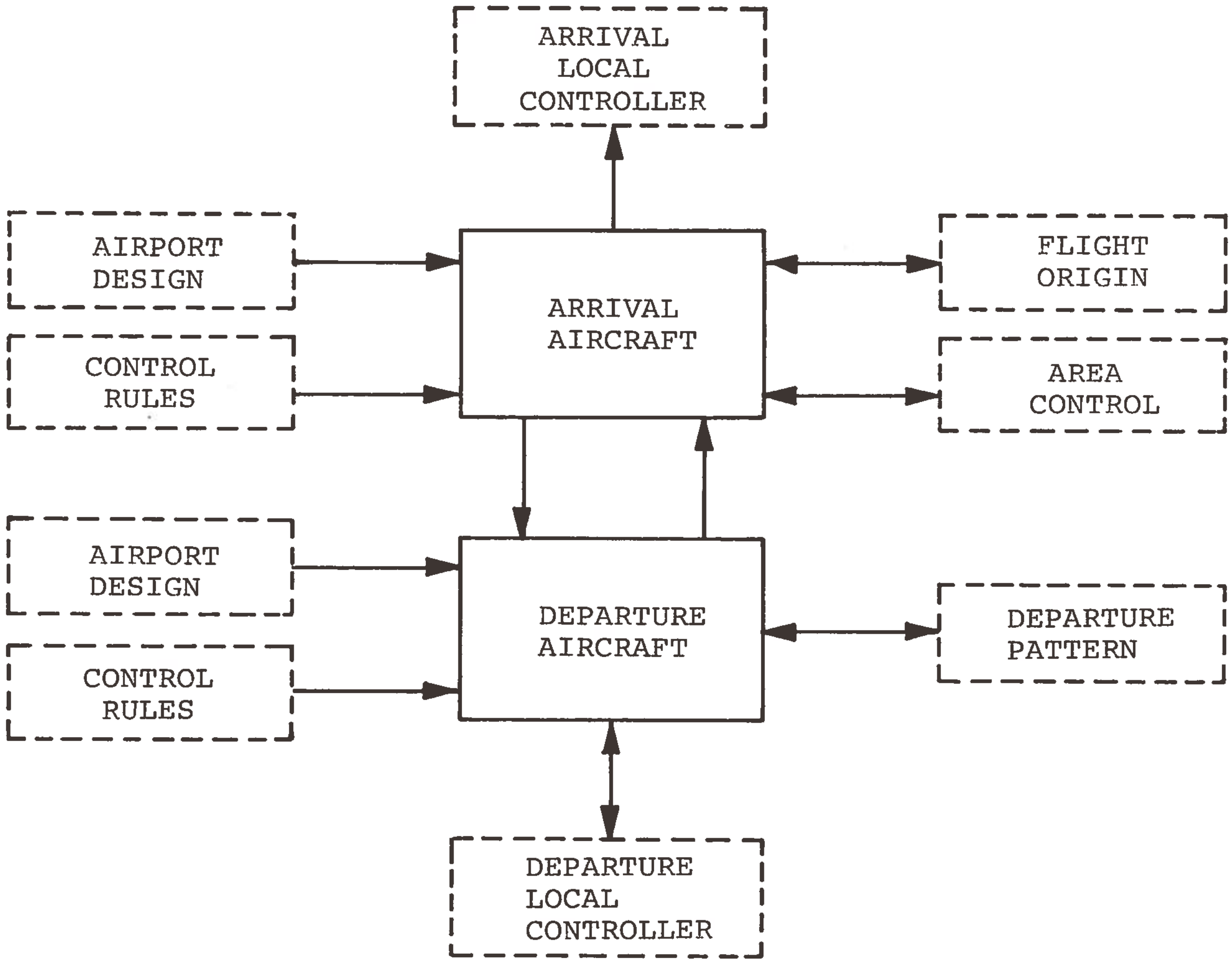

Figure 3. Radar Controller

to resolve potential and actual conflicts. A potential conflict is defined as two aircraft less than five miles apart whose point of closest approach is a conflict. (However, aircraft on a localizer path are not considered to be in potential conflict.) In the event of a potential conflict the current rules are:

1. If two aircraft are not co-altitude, then the aircraft are assigned altitudes to maintain altitude separation or to increase altitude separation.

2. Aircraft at the same altitude are turned from their current course to increase separation at the point of closest approach.

DEPARTURE AIRCRAFT MODULE

An aircraft is handed off from the Departure Local Controller and issued vectors until it is five miles away from the airport. At this point the program routes the aircraft along a standard instrument departure (SID). Handoff from the model occurs when an aircraft is more than thirty miles from the airport center and on, or near, a SID path.

LOCAL CONTROLLER

The Local Controller portion of the model (Figure 4) encompasses three controller functions. The first is the Ground Controller which monitors runway assignments and determines when a runway is available. This controller also monitors the landing and departure queues, and assigns aircraft in the queues to runways. The second function is the Arrival Local Controller which controls arriving aircraft from the time they are handed off from the Arrival Aircraft Module until they enter the taxiway. The third controller function is the Departure Local Controller which controls departing aircraft from the time they enter the runway line-up until they are airborne and handed off to the Departure Aircraft Module.

GROUND CONTROLLER MODULE

The Ground Controller Module controls departing aircraft from the gate until they are in the runway line-up on the taxiway. Arriving aircraft are controlled from runway touch-down until the time they arrive at the gate.

The procedure for departing aircraft is to check the departure schedule at each step to determine if an aircraft departure is scheduled. If none is scheduled, the program proceeds to the next step. If a departure is scheduled, the aircraft is assigned to the departure runway handling that type of aircraft which has the smallest queue. Thus, it is possible to specify that STOL aircraft use STOL runways, and other similar restrictions.

Next, the Ground Control Module vectors ground aircraft to their runway queue, allowing them to cross available runways. (A runway is available if no arriving aircraft is within the two miles of landing on that runway, or a dependent runway, and if no other departing aircraft has been given takeoff clearance.)

Departure and arrival runways are designated. In cases where parallel runways are used, one is designated for departure and one for arrival, and assignments are made to each runway on a first come first served basis. When a single runway is used for both departure and arrival, arrivals are given preference, and the ratio of arrivals to departures is a function of mean waiting time and maximum waiting time for aircraft in the arrival queue. The exact function is an input. In this version of the program there are no provisions for emergency landings.

AIRCRAFT ARRIVAL

The Aircraft Arrival portion of the model (Figure 5) is divided into four modules. The Flight Origin module simulates handoffs from the enroute controller to the Area Control module. In turn, the Area Control module assigns controllers and resets controller rules if the area has special rules. The Approach Pattern module handles aircraft motion and communication in response to the Arrival Local Controller. The Taxiway and Flight Termination module moves the aircraft from the landing point to the gate.

FLIGHT ORIGIN MODULE

The Flight Origin module regulates controller workload by limiting acceptance rate (i.e., number of aircraft per controller). It checks the arrival schedule, and if an arrival is scheduled, it sets up a point of handoff and the velocity, descent rate, and bearing.

AREA CONTROL MODULE

The Area Control module simulates the aircraft response to the vectors generated in the Arrival Aircraft module. The aircraft position is updated using the latest vector as a basis and then correcting for wind error and aircraft characteristics. The wind error in this program is a random number of limited magnitude (set by the user) which is added to the position. The aircraft characteristics which limit response to a vector are maximum climb rate, maximum descent rate, maximum and minimum velocity and maximum turn rate.

The module simulates communications between aircraft and controller by adding random errors to the position and velocity data before transmitting this information to the controller by voice, radar, or transponder mode C.

APPROACH PATTERN MODULE

The Approach Pattern module governs aircraft response to vectors from the Arrival Local Controller module and handles communications with that module. The major difference between this module and the Area Control module is in the rules applied (i.e., communications are more frequent, radar is more accurate, and for each aircraft type there are rules covering altitude in relation to speed).

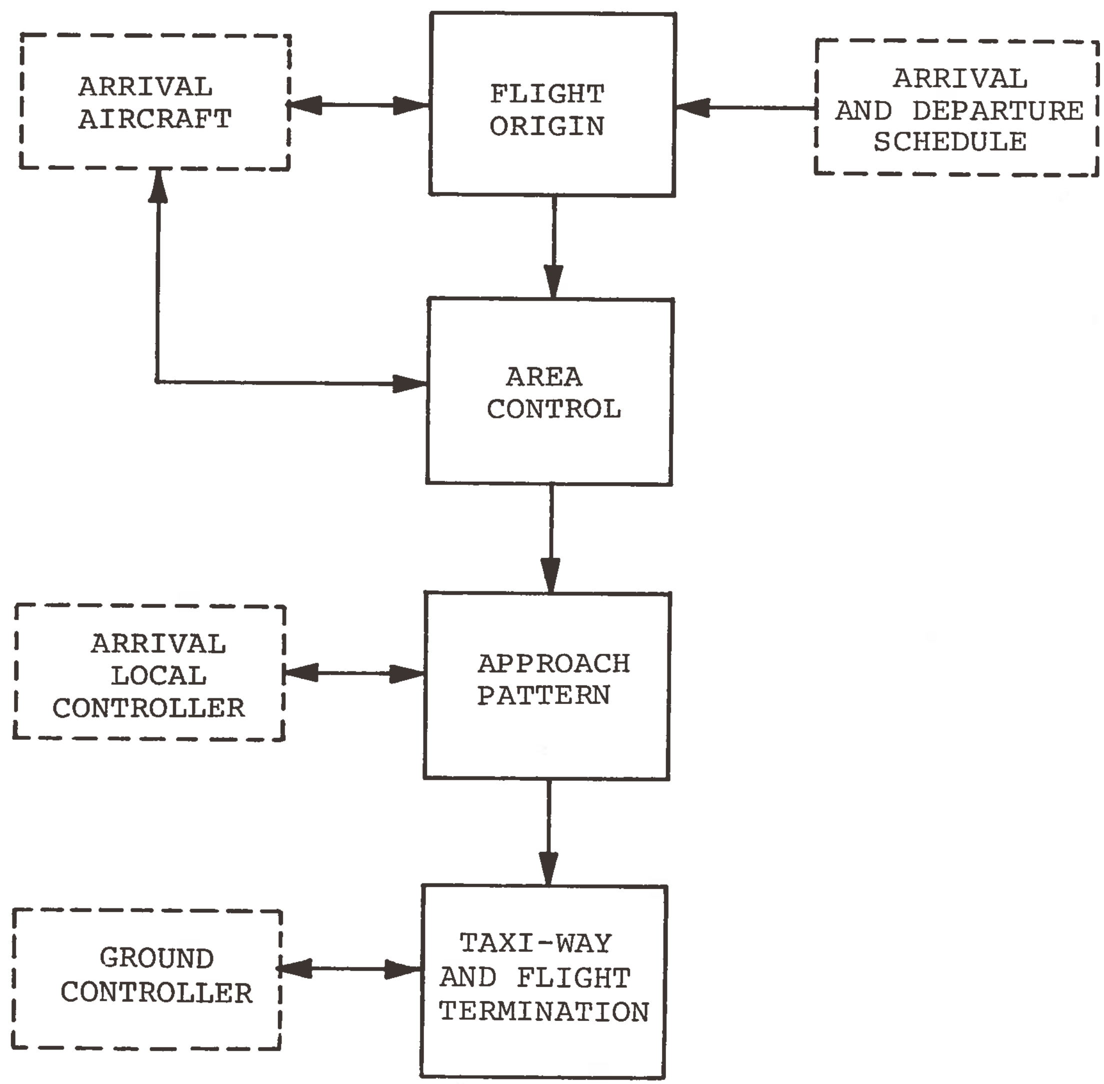

Figure 5. Aircraft Arrival

TAXIWAY AND FLIGHT TERMINATION MODULE

The Taxiway and Flight Termination module receives a gate assignment from the Ground Controller module. It then simulates the aircraft following a taxiway map, moving from node to node towards its gate. The aircraft are processed on a first come first served basis and each is moved in such a manner that no two aircraft cross the same intersection at the same time. Aircraft move at 15 to 20 knots and are separated by no less than one plane length at all times. The prevailing rule is see and be seen.

AIRCRAFT DEPARTURE

The Aircraft Departure portion of the model contains three modules which execute aircraft operations (i.e., communications and aircraft movements). The Taxiway and Flight Start module processes aircraft movement from gate to taxiway line up. The Departure Local Control module processes aircraft movement from clearance to take-off until the runway is available for the next aircraft. The Departure Pattern module processes aircraft motion from this point until it leaves the airport area.

TAXIWAY AND FLIGHT START MODULE

This module contains a list of departing aircraft. When the schedule calls for a departure, the aircraft is assigned to a gate, and the Ground Controller module is called upon to assign the aircraft to a runway. The Taxiway and Flight Start module then moves the aircraft along a fixed path to the runway, resolving conflicts between crossing aircraft on a first come first served basis. When an aircraft is about to cross a runway the aircraft is stopped and a message transmitted to the Ground Controller module which transmits either a wait order or a runway clear order. After receiving a runway clear order the aircraft crosses the runway and proceeds to the end of the runway line-up.

DEPARTURE AIRCRAFT CONTROL MODULE

This module receives a "runway cleared for departure" message from the Departure Local Controller module. It then brings the first aircraft onto the runway. The Departure Local Controller module is notified; it clears the aircraft for take-off; introduces a time delay to simulate take-off; and sets the aircraft take-off velocity and climb angle. The module then records position and transmits it through a radar simulator to the Departure Local Controller module. It also transmits, on request, data by voice communication. It receives vectors from the Departure Local Control module and executes them as fast as the aircraft characteristics permit. Finally, aircraft position is updated at regular intervals by this module.

DEPARTURE PATTERN MODULE

This module models aircraft motion under a departure radar controller. It provides voice communication and radar blips to the Departure Aircraft module. It receives vectors from the Departure Aircraft module and models aircraft response to these vectors. Its principal algorithm checks vectors against aircraft capability and acts accordingly.

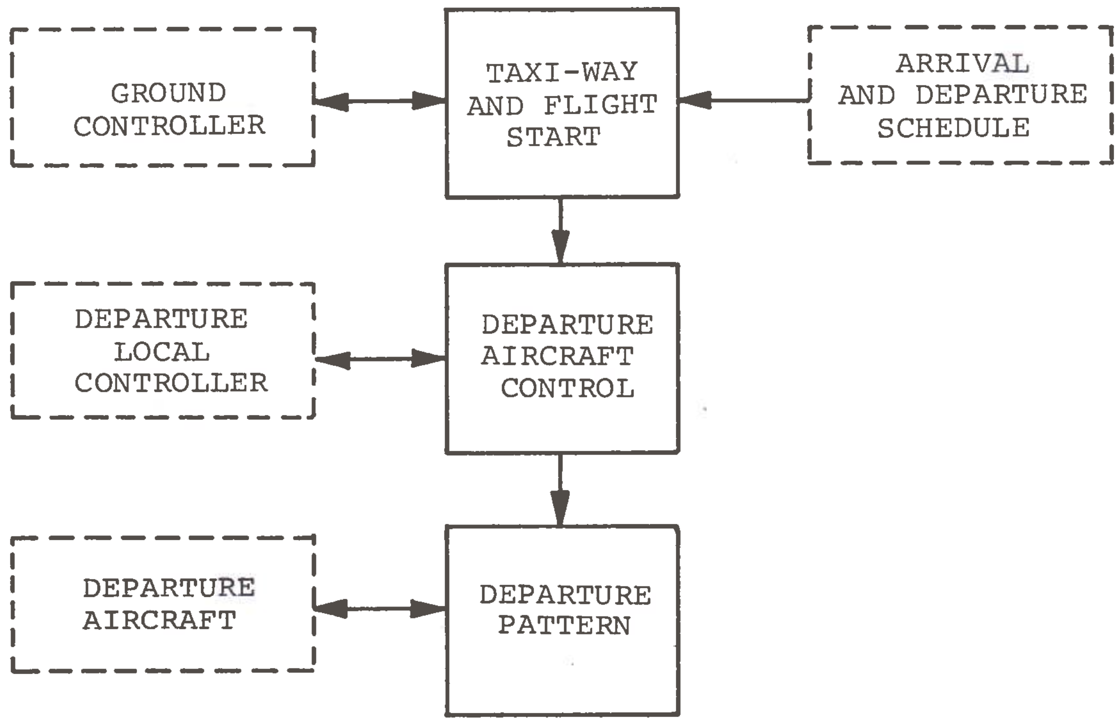

Figure 6. Aircraft Departure

It also adds an error to the aircraft bearing and velocity set by the pilot in response to the departure radar controller's direction. These are generated from a random distribution with a maximum value set at a fixed percent of the actual value. However, the distribution used can be changed by the user. The wind correction is set by moving the aircraft position by adding random numbers to the aircraft position. The maximum random number is set by the user.

REFERENCES

1. Flenner, W., Terminal Air Traffic Control, 7110.8B, Federal Aviation Administration, April 1971.

2. Report of Department of Transportation, Air Traffic Control Advisory Committee, December 1969.

3. Bierman, R., Airport Surface Control and Guidance Systems Development Plan (Draft), Federal Aviation Administration.

4. Inter-metropolitan STOL Evaluation, American Airlines. January 1970.

5. Horonjeff, R., Planning and Design of Airports, McGraw-Hill, 1962.

6. deHollan, A. N., and Priver, A. S., A Conceptual Network Model of the Air Transportation System, Report No.DOT-TSC-FAA-71-3, Department of Transportation, April 1971.

7. Ruetenik, J. R., and Lang, W., Analysis of PWI Performance in Terminal Area Traffic, Report RA TR-67, Kaman AviDyne.

8. Mengert, P. H., and Englander, I., Index of Orderliness Statistical Validation on ARTS III Test Data, Department of Transportation, April 1971.

APPENDIX A
Simulation Program Listings

```
$JOB  I13710 2 TCD      ENGLANDER                         001 020
$EXECUTE        IBJOB
$IBJOB
$IBFTC ONE
C OSO4 AIRPORT CAPACITY JUNE 14,1971
C IDO ARE RUNWAY  ASSIGNMENT POSITIVE ARRIVE, NEGATIVE DEPARTURE
      COMMON/A/ISENSE
      COMMON /M/MTRAN
      COMMON /R/ IDO
      COMMON/P/ XS,YS,ZS,T
      COMMON/U/DT,TI
      COMMON/B/COUNT
      COMMON/C/ZMAX,ZMIN,TMAX,TMIN
      DIMENSION MTRAN(144)
      DIMENSION IDIST(100),JDIST(100)
      DIMENSION ZMAX(30),ZMIN(30),TMAX(30),TMIN(30)
      DIMENSION PRI(4)
      DIMENSIONIDENT(4,30),JTYPE(4,30)
      DIMENSION IDO(4),KS(4)
      DIMENSION TOC(4),TC(4)
      DIMENSION PH(4),XS(4),YS(4),ZS(4),S(4),TS(4),IS(4),T(4),V(4),P(4)
      DIMENSION COUNT(10)
      DIMENSION X(4,30),Y(4,30),Z(4,30),TT(4,30),VT(4,30),PT(4,30)
      DIMENSION XA(4,30),YA(4,30),ZA(4,30),TD(4,30),VD(4,30),PD(4,30)
      DIMENSION ISENSE(4)
      DIMENSION   CCOUN(5,9)
      IID=0
  996 CONTINUE
      IF(IID.LE.0) GO TO 997
      DO999 I=1,9
      CCOUN(IID,I)=COUNT(I)
  999 CONTINUE
      MTRAN(1)= -9*IID
      K=1
      DO 998 KK=1,IID
      DO 998 LL=1,9
      K=K+1
      MTRAN(K)=CCOUN(KK,LL)
 998  CONTINUE
      CALL CPSXN(IWORD,MTRAN,JUNN)
  997 CONTINUE
      IID=IID+1
C N IS NUMBER OF RUNWAYS
            CALL DATA(N)
      PI=3.14159
      KSIGN=0
      JRXUA=1
      DO 6 IJKU=1,144
      MTRAN(IJKU)=0
    6 CONTINUE
C           CALL INNERT
C VEHICLE ACCESS
C****************
C****************
C****************
C****************
      JUNK=0
  209 FORMAT (6F10.5,2I5)
C OUTPUT CONTROL SECTION
```

```
  LABERT ST LOUIS SPECIAL
        LAMBER=-1
 4001 CONTINUE
C TWRITE IS INNITIAL PRINTING OF PLANE TABLE
        TWRITE=0.0
C TLIST CONTROLS CONFLICT TABLE PRINTING
        TLIST=0.
C DTWRIT IS WRITING INTERVAL
        DTWRIT=10.
  102 FORMAT(1X,F9.5,5F10.5)
C SET RESULTS COUNTERS TO INNITIAL VALUES
        DO 462 I=1,10
        COUNT(I)=0.0
  462 CONTINUE
C DT IS TIME INCREMENT
        DT=0.01
C ITYPE IS THE NUMBER OF AIRCRAFT TYPES
        ITYPE=2
        NCLASS=2*ITYPE
        MTRAN(1)=N
C ISENSE ARE CONTROLS OF INTERACTIVE INPUT
        ISENSE(1)=0
        ISENSE(2)=0
        ISENSE(3)=0
        ISENSE(4)=0
        DO 300 J=1,N
        ISENSE(J)=1
  300 CONTINUE
C M IS MAXIMUM NUMBER OF AIRCRAFT
        M=30
C TI IS PROBLEM TIME
        TI=-DT
C INPUT SECTION
C PH IS INTERRUNWAY PHASE SHIFT FOR ARRIVALS
        READ(5,102)(PH(IFF),IFF=1,NCLASS)
C S IS AIRCRAFT DISPLACEMET ON ARRIVAL IN AIRPORT AREA
        READ(5,102)(S (IFF),IFF=1,NCLASS)
C       WRITE(6,102)(PH(IFF),IFF=1,NCLASS)
C       WRITE(6,102)(S (IFF),IFF=1,NCLASS)
C TOC IS MINIMUM OCCUPANCY TIME
        TOC(1)=1.
        TOC(2)=1.
        TOC(3)=1.
        TOC(4)=1.
        DO 767 J=1,N
C INPUT NEW PLANE CHARACTERISTICS. POSITION,BEARING,VELOCITY,CLIMB
        READ(5,102)XS(J),YS(J),ZS(J),T(J),V(J),P(J)
C       WRITE(6,102)XS(J),YS(J),ZS(J),T(J),V(J),P(J)
        V(J)=V(J)/60.
  767 CONTINUE
C INPUT AIRCRAFT CHARACTERISTICS
C ZMAX MAXIMUM CLIMB RATE FT./HR.
C ZMIN MAXIMUM DECENT RATE
C PDMAX MAXIUM ACCENT ANGLE
C PDMIN MAXIUM DECENT ANGLE
C TMAX AND TMIN MAXIMUM TURN ANGLES.
        DO 127 IT=1,ITYPE
        READ(5,102) ZMAX(IT),ZMIN(IT),TMAX(IT)
C       WRITE(6,102)ZMAX(IT),ZMIN(IT),TMAX(IT)
```

```
      TMIN(IT)=-TMAX(IT)
  127 CONTINUE
  100 FORMAT(4I1)
C INNNITIALIZE
      DO1J=1,N
C IS IS AIRCRAFT ASSIGNED TO RUNWAY
      IS(J)=0
C KS(J) IS QURUE COUNT
      KS(J)=0
C TC IS TIME RUNWAY OCCUPIED
      TC(J)=0.
C TS(J) IS TIME SINCE LAST ARRIVAL
      TS(J)=0.
      DO 1I=1,M
C X,Y,Z IS AIRCRAFT POSITION
      X(J,I)=0.
      Y(J,I)=0.
      Z(J,I)=0.
      IDENT(J,I)=0
      JTYPE(J,I)=J/3 +1
    1 CONTINUE
C INCREMENT PROBLEM TIME
    7 TI= TI +DT
      IF(TI.GT.30.) GO TO 996
C     CALL VACS
C VEHICLE ACCESS
C *****************
C *****************
C *****************
C *****************
      IF(TI.LT.TWRITE) GO TO 1101
C PRINT OUTPUT FOR PROBLEM INTERVAL
      TWRITE=TWRITE+DTWRIT
C     WRITE(6,902)(COUNT(L),L=1,9) ,(KS(L),L=1,4)
C RESET INDEX OF ORDERLINESS
      COUNT(7)=0.
  902 FORMAT(9F10.4,4I5)
C LABERT ST LOUIS SPECIAL
 1101 CONTINUE
      IAIR=0
      PRIOR=0.0
      DO797 J=1,N
      IAIR=IS(J)+IAIR
      PRI(J)=(TI/S(J))-COUNT(J)
      IF(IDO(J).GT.0) PRIOR=PRIOR+PRI(J)
  797 CONTINUE
      IF(JUNK.GT.2) MTRAN(1)=IAIR
      ITRAN=1
      DO 750 J=1,N
      IIS=IS(J)
      XIS=IIS
      DO 502 I=1,IIS
  505 FORMAT(5E20.8)
      IXAIR=X(J,I)*1500./40.
      IF(JUNK.LT.1) GO TO 502
      JFIF=2000.
      IFIF=-2000
      IF(IXAIR.GT.JFIF.OR.IXAIR.LT.IFIF) MTRAN(1)=MTRAN(1)-
      IF(IXAIR.GT.JFIF.OR.IXAIR.LT.IFIF) GO TO 502
```

```
      ITRAN=ITRAN+1
      IF(ITRAN.GT.144)GO TO 502
      MTRAN(ITRAN)=IXAIR
      IYAIR=Y(J,I)*1500./40.
      IF(IYAIR.GT.JFIF.OR.IYAIR.LT.IFIF) ITRAN=ITRAN-1
      IF(IYAIR.GT.JFIF.OR.IYAIR.LT.IFIF) MTRAN(1)=MTRAN(1)-1
      IF(IYAIR.GT.JFIF.OR.IYAIR.LT.IFIF) GO TO 502
      ITRAN=ITRAN+1
      IF(ITRAN.GT.144)GO TO 502
      MTRAN(ITRAN)=IYAIR
      IZAIR=Z(J,I)*3000./8000.        -2000.
      IF(IZAIR.GT.JFIF.OR.IZAIR.LT.IFIF) ITRAN=ITRAN-2
      IF(IZAIR.GT.JFIF.OR.IZAIR.LT.IFIF) MTRAN(1)=MTRAN(1)-1
      IF(IZAIR.GT.JFIF.OR.IZAIR.LT.IFIF) GO TO 502
      ITRAN=ITRAN+1
      IF(ITRAN.GT.144) GO TO 502
      MTRAN(ITRAN)=IZAIR
      ITRAN=ITRAN+1
      IF(ITRAN.GT.144) GO TO 502
      MTRAN(ITRAN)=IDENT(J,I)
  502 CONTINUE
  750 CONTINUE
      IWORD=2*IAIR+1
      IF(ISENSE(1).GT.0.OR.ISENSE(2).GT.0) IWORD=0
      IF(ISENSE(3).GT.0.OR.ISENSE(4).GT.0) IWORD=0
      IF(IWORD.LE.0) GO TO 10
      JUNN=0
      IWORD=144
    8 FORMAT(1X,I10)
      IF(MTRAN(1).LE.0) GO TO 298
      CALL CPXSN(IWORD,MTRAN,JUNN)
C
C
C
C
  298 CONTINUE
C*********************
      JUNK=5
   10 CONTINUE
C     WRITE(6,308)
      NNJ=N -1
  308 FORMAT(15H CONFLICT TABLE)
      JRXUA=JRXUA+1
      IF(JRXUA.GE.N) JRXUA=1
      J=JRXUA
      IIS=IS(J)
C FLOW CONTROL
      IFLOW=10
      IF(IIS.GT.IFLOW) S(J)=S(J)*1.10
        IF(IIS.LE.0) S(J)=0.90*S(J)
      IF(IIS.LT.1) GO TO 1103
C CONFLICT PARAMETERS
C DISTR ACTION POINT FOR CONTROLLER BASED ON PROJECT STUPID
      DISTR=5.0
C DISTL MINIMUM SEPERATION
      DISTL=3.0
C ZCON MINIMUM ALTITUDE
      ZCON =800.
C ALTM MINIMUM ALTITUDE SEPERATION
```

```
      ALTM=900.
      DO 301 K=J,N
      LLS=IS(K)
      IF(LLS.LT.1) GO TO 301
      DO 302 I=1,IIS
      IF(Z(J,I).LT.ZCON) GO TO 302
      DO 303 L=1,LLS
      IF(Z(K,L).LT.ZCON) GO TO 303
C PREVENT ARRIVAL DEPARTURE INTERACTION
      ZCOM=3200.
      IF(Z(J,I).LT.ZCOM.AND.IDO(J).NE.IDO(K)) GO TO303
C AIRCRACT UNDER LOCAL CONTROLLEP
      ZCOMM=2200.
      IF(Z(J,I).LT.ZCOMM.AND.IDO(J).GT.0) GO TO 303
      IF(Z(K,L).LT.ZCOMM.AND.IDO(K).GT.0) GO TO 303
      IF(Z(K,L).LT.ZCOM.AND.IDO(J).NE.IDO(K)) GO TO303
      ALT=ABS(Z(J,I)-Z(K,L))
      IF(ALT.GT.ALTM) GO TO 303
      DIST =SQRT(((X(J,I)-X(K,L))**2)+((Y(J,I)-Y(K,L))**2))
      IF(DIST.GT.DISTR) GO TO 303
C CHECK FOR POSITIVE RELATIVE VELOCITY
      KDIST=20.*DIST
      IF(KDIST.GT.100) KDIST=100
      IF(KDIST.LT.1) KDIST=1
      IDIST(KDIST)=IDENT(J,I)
      JDIST(KDIST)=IDENT(K,L)
      LDIST=KDIST-1
      IF(LDIST.LE.0) GO TO 970
      DO 982 IRS=1,LDIST
      IF(IDENT(J,I).EQ.IDIST(IRS).AND.IDENT(K,L).EQ.JDIST(IRS))GO TO 970
  982 CONTINUE
      GO TO 303
  307 FORMAT(4I5,2F10.5)
  970 IF(K.EQ.J.AND.L.LE.I) GO TO 303
C COUNT(7) IS INDEX OF ORDERLINESS
      XMU=IS(J)*IS(K)
      XMU=XMU*(DIST**2)
      IF(XMU.LT.1.E-05) XMU=1.E-05
      COUNT(7)=COUNT(7)+(1./XMU)
C LABERT ST LOUIS SPECIAL
      IF( ILIST.GT.II) GO TO 1102
C     WRITE(6,307) J,J,IDENT(J,I),IDENT(K,L),ALT,DIST
 1102 CONTINUE
      IF(DIST.GT.DISTL) GO TO 980
C COUNT(6) IS THE CONFLICT COUNTER
      COUNT(6)=COUNT(6)+DT
      IF(DIST.LT.1.E-05) DIST=1.E-05
      COUNT(8)=COUNT(8)+(DT/DIST)
C LABERT ST LOUIS SPECIAL
      GO TO 980
      IF(ILIST.GT.II) GO TO 980
C     WRITE(6,307) J,IDENT(J,I),K,IDENT(K,L),ALT,DIST
C HORIZONTAL CONFLICT RESOLUTION MAKES THE AIRCRAFT PATHS MORE PARALLEL
C DELETE CON+LICT RESOLUTION THIS RUN
  980 GO TO 303
C 980 IT=JTYPE(J,I)
      COR=PI/2.
      IF(COS(TD(J,I)).EQ.0.) GO TO 762
      XDIF=((XA(J,I)-XA(K,L))/COS(TD(J,I)))
```

```
      TCOS=COS(TD(K,L))/COS(TD(J,I))
  762 TDX=TD(K,L)-COR-TD(J,I)
      IF(YA(J,I).GT.YA(K,L)) TDX=-TDX
      IF(XDIF.GT.0.) TDX=0.0
      IF(TCOS.GT.0.) GO TO 802
      TDX=0.0
      IF(COS(TD(J,I)).LT.0..AND.XA(J,I).LT.XA(K,L)) GO TO 802
      IF(COS(TD(J,I)).GT.0..AND.XA(J,I).GT.XA(K,L)) GO TO 802
      IF(YA(K,L).EQ.YA(J,I)) YA(J,I)=YA(J,I)+1.E-05
      COR=((YA(K,L)-YA(J,I))/ABS((YA(K,L)-YA(J,I))))*COR
      IF(COS(TD(K,L)).LT.0.) COR=-COR
      TDX=COR
  802 CONTINUE
      TXX=TMAX(IT)*DT
      TNN=TMIN(IT)*DT
      IF(TDX.GT.PI) TDX=TDX-2.*PI
      PIJ=-PI
      IF(TDX.LT.PIJ) TDX=2.*PI-TDX
      IF(TDX.GT.TXX) TDX=TXX
      IF(TDX.LT.TNN) TDX=TNN
      TD(J,I)=TD(J,I)+TDX
      IT=JTYPE(K,L)
      COR=PI/2.
      IF(COS(TD(J,I)).EQ.0.OR.COS(TD(K,L)).EQ.0.) GO TO 9999
      XDIF=((XA(K,L)-XA(J,I))/COS(TD(K,L)))
      TCOS=COS(TD(K,L))/COS(TD(J,I))
 9999 CONTINUE
      TDX=TD(J,I)-COR-TD(K,L)
      IF(YA(K,L).GT.YA(J,I)) TDX=-TDX
      IF(XDIF.GT.0.) TDX=0.0
      IF(TCOS.GT.0.) GO TO 803
      TDX=0.0
      IF(COS(TD(K,L)).LT.0..AND.XA(K,L).LT.XA(J,I)) GO TO 803
      IF(COS(TD(K,L)).GT.0..AND.XA(K,L).GT.XA(J,I)) GO TO 803
      IF(YA(J,I).EQ.YA(K,L)) GO TO 803
      COR=((YA(J,I)-YA(K,L))/ABS((YA(K,L)-YA(J,I))))*COR
      IF(COS(TD(J,I)).LT.0.) COR=-COR
      TDX=COR
  803 CONTINUE
      TXX=TMAX(IT)*DT
      TNN=TMIN(IT)*DT
      IF(TDX.GT.PI) TDX=TDX-2.*PI
      PIJ=-PI
      IF(TDX.LT.PIJ) TDX=2.*PI-TDX
      IF(TDX.GT.TXX) TDX=TXX
      IF(TDX.LT.TNN) TDX=TNN
      TD(K,L)=TD(K,L)+TDX
C VERTICAL CONFLICT RESOLUTION MAKES THE AIRCRAFT PATHS SLOW
C COALTITUDE APPROACH
C XEX IS MAXIMUM DECENT ANGLE FOR EMERGENCIES
      XEX=2.*PI*DT/180.
      EX=(ZA(J,I)-ZA(K,L))/ALTM
      IF(EX.EQ.0.) EX=1./XEX
      EX=1./EX
      IF(EX.GT.XEX) EX=XEX
      XEX=-XEX
      IF(EX.LT.XEX) EX=XEX
      PD(J,I)=PD(J,I)-EX
      PD(K,L)=PD(K,L)+EX
```

```
      CALL VELO(TD(J,I),TT(J,I),VD(J,I),VT(J,I),PD(J,I),PT(J,I))
      CALL VELO(TD(K,L),TT(K,L),VD(K,L),VT(K,L),PD(K,L),PT(K,L))
  303 CONTINUE
  302 CONTINUE
  301 CONTINUE
      IF(TLIST.GT.TI) GO TO 1103
      TLIST=TLIST+0.1
 1103 CONTINUE
  103 DO2KJ=1,NCLASS
      J=KJ
      IF(NCLASS.GT.N) J=KJ-2
C CHECK FOR NEW PLANE ARRIVAL
C HAND OFF FROM ON ROUTE CONTROLLER
      IF(TI.GT.PH(KJ)) TS(KJ)=TS(KJ)+DT
      IF(TS(KJ).LT.S(KJ)) GO TO 2
      TS(KJ)=0.
C VACS INSETRS
C ****************
C ****************
C ****************
C     IF(IDO(J).LT.0) CALL VACB(J,KT)
C     IF(IDO(J).LT.0.AND KT.LT.0) GO TO 2
C UPDATE PLANE COUNTER FOR NEW ARRIVAL
      IF(IDO(J).GT.0)IS(J)=IS(J)+1
      IF(IS(J).GT.M) IS(J)=M
C SET COUNTER FOR NEW DEPARTURE QUEUE ENTRY
      IF(IDO(J).LT.0) KS(J)=KS(J)+1
      IF(IDO(J).GT.0) I=IS(J)
      IF(IDO(J).LT.0) I=KS(J)+IS(J)
      IF(I.GT.M)I=M
      KSIGN=KSIGN+1
      IF(KSIGN.GT.99) KSIGN=1
      IDENT(J,I)=KSIGN
      JTYPE(J,I)=KJ/3 +1
      IF(JTYPE(J,I).GT.ITYPE) JTYPE(J,I)=ITYPE
C TRANSFER POSITION TO PILOT
  104 CALL ARIVIT(J)
      CALL POSIT(XS(J),X (J,I),YS(J),Y (J,I),ZS(J),Z(J,I))
C TRANSFER POSITION TO CONTROLLER VIA RADAR ETC.
      CALL POSIT(XS(J),XA(J,I),YS(J),YA(J,I),ZS(J),ZA(J,I))
C TRANSMIT VELOCITY,BEARING,CLIMB TO PILOT VIA INSTRUMENTS
      CALL VELO(T(J),TT(J,I),V(J),VT(J,I),P(J),PT(J,I))
C TRANSMIT MOTION DATA TO CONTROLLER VIA DATA LINK
      CALL VELO(T(J),TD(J,I),V(J),VD(J,I),P(J),PD(J,I))
    2 CONTINUE
   12 FORMAT(1X,I10,I10)
   11 FORMAT(1X,I10,F10.5)
      J=1
  375 CONTINUE
      I=0
C CHECK FOR TAKEOFF
      IF(IDO(J).LT.0) GO TO 205
   17 IF(IS(J).LT.1) GO TO 3
      I=I+1
   13 FORMAT (1X,6F10.5)
      XPI=2.*PI
      IF(TT(J,I).GT.PI) TT(J,I)=TT(J,I)-XPI
        PI=-PI
      IF(TT(J,I).LT.PI) TT(J,I)=TT(J,I)+XPI
```

```
      PI=3.14159
      CALL PLANE(X(J,I),Y(J,I),Z(J,I),TT(J,I),VT(J,I),PT(J,I))
C CONTROLLNR-ESTIMATES NEW POSITION AND UPDATES INSTRUCTIONS IF REQUIRED
C COUNT(10) CONTROLS STACKING OF EXCESS ARRIVAL AIRCRAFT
      COUNT(10)=I*IDO(J)
      CALL PLAND(XA(J,I),YA(J,I),ZA(J,I),TD(J,I),VD(J,I),PD(J,I),J,
     1JTYPE(J,I),I)
C TRANSMIT POSITION TO CONTROLLER VIA RADAR ETC.
            CALL POSIT(X(J,I),XA(J,I),Y(J,I),YA(J,I),Z(J,I),ZA(J,I))
C CONTROLLER TRANSMITS DIRECTIONS TO PILOT
            CALL  VELO(TD(J,I),TT(J,I),VD(J,I),VT(J,I),PD(J,I),PT(J,I))
C GO TO 200 FOR DEPARTING AIRCRAFT TAKE OFF ROUTINE
      IF(IDO(J).LT.0) GO TO 200
C ARRIVING AIRCRAFT LANDING ROUTINE
C INTER RUNWAY CONFLICT ARRIVAL CONTROL
      NDEP=-2
      JDEP=ISENSE(J)
      IF(ISENSE(J).LT.0 )ISENSE(J)=0
      IF(JDEP.LT.NDEP)  TS(J)=TS(J)-DT
C IF PLANE HAS NOT LANDED GO TO 5
      IF(Z(J,I).GT.0.) GO TO 5
C VAXC INSERT
C **************
C **************
C **************
C **************
C     CALL VACX(X(J,I),Y(J,I),J,IDENT(J,I))
C COMPUTE LANDING COUNTERS AND OCCUPANCY
C COUNT(5) IS DUAL OCCUPANCY COUNTER
            CALL DUAL(TC,J)
      IF(TC(J).LE.TI)TC(J)=0.
      IF(TC(J).GT.0.)COUNT(5)=COUNT(5)+1.
      TC(J)=TI + TOC(J)
C COUNT(J) IS NUMBER OF PLANES LANDED
      COUNT(J)=COUNT(J)+1.
C DELETE LANDED PLANE FROM CONTROLLER WORK LOAD
      ISJ=IS(J)
      ISI=I+1
      IF(ISI.GE.ISJ)GO TO 270
      DO 4 K=ISI,ISJ
      X(J,K-1)=X(J,K)
      Y(J,K-1)=Y(J,K)
      Z(J,K-1)=Z(J,K)
      TT(J,K-1)=TT(J,K)
      VT(J,K-1)=VT(J,K)
      PT(J,K-1)=PT(J,K)
      IDENT(J,K-1)=IDENT(J,K)
    4 CONTINUE
  270 CONTINUE
      IS(J)=IS(J)-1
      I=I-1
      IF(IS(J).GT.M) IS(J)=M
    5 IF(I.LT.IS(J)) GO TO 17
      GO TO 3
C DEPARTURE ROUTINE
  200 DMAX=40.
      XOI=0.0
      YOI=0.0
      DRUN=SQRT((((X(J,I)-XOI)**2)+((Y(J,I)-YOI)**2)))
```

```
      IF(DRUN.LT.DMAX)GO TO 5
C     HAND OFF REQUIRED
      XH=X(J,I)
      YH=Y(J,I)
      ZH=Z(J,I)
      TH=TT(J,I)
      VH=VT(J,I)
      PHH=PT(J,I)
      JDENT=IDENT(J,I)
      II=IS(J)-1
      IF(II.LT.I) GO TO 275
      DO 204 K=I,II
      X(J,K)=X(J,K+1)
      Y(J,K)=Y(J,K+1)
      Z (J,K)=Z (J,K+1)
      TT(J,K)=TT(J,K+1)
      VT(J,K)=VT(J,K+1)
      PT(J,K)=PT(J,K+1)
      IDENT(J,I)=IDENT(J,K+1)
  204 CONTINUE
C REDUCE CONTROLLER LOAD
  275 IS(J)=IS(J)-1
      I=I-1
      GO TO 5
C RUNWAY OCCUPIED
  205         CALL DUAL(TC,J)
      IF(TC(J).GT.TI) GO TO 17
      TC(J)=0.0
C                        IF BACKUP IN LANDING SUSPEND TAKEOFFS
      IF(IDO(J).LT.O.AND.PRIOR.GT.10.)GO TO 17
C NO AIRCRAFT READY FOR TAKEOFF GO TO 17
      IF(KS(J).LE.0) GO TO 17
C CHECK FOR CONFLICT WITH DEPENDENT RUNWAY
      NDEP=-2
      JDEP=ISENSE(J)
      IF(ISENSE(J).LT.0) ISENSE(J)=0
      IF(JDEP.LT.NDEP) GO TO 17
C COUNT TAKEOFFS
      COUNT(J)=COUNT(J)+1.
      TC(J)=TI+TOC(J)
C REDUCE QUEUE
      KS(J)=KS(J)-1
C INCREASE CONTROLLER LOAD
      IS(J)=IS(J)+1
      IF(IS(J).GT.M)IS(J)=M
      IIA=IS(J)
C VTAC IS TAKEOFF VELOCITY
C
C
C     STOL CORRECTION
      IF(J.LT.3) VTAC=140./60.
      IF(J.GE.3) VTAC= 70./60.
C
C
      VD(J,IIA)=VTAC
      VT(J,IIA)=VTAC
C
C
C STOL CORRECTION
```

```
      PDTAC=-3.*PI/180.
      IF(J.GE.3) PDTAC=-8.*PI/180.
      PT(J,IIA)=PDTAC
      PD(J,IIA)=PDTAC
      IF(KS(J).GT.0) TC(J)=TI+TOC(J)
 4000 FORMAT(1X,20HERROR ISTEP ALTITUDE,E20.9,3I10)
      GO TO 17
C   3 CONTINUE
    3 J=J+1
      IF(J.LE.N) GO TO 375
      GO TO 7
      END
$IBFTC TWO
      SUBROUTINE STACK(ITC,XA,YA,ZA,TD)
      DIMENSION XO(4),YO(4)    ,RR(4)
      PI=3.14159/2.
      RR(1)=20.
      RR(2)=20.
      RR(3)=20.
      RR(4)=20.
      XO(1)=20.
      XO(2)=0.
      XO(3)=-20.
      XO(4)=0.
      YO(1)=0.
      YO(2)=20.
      YO(3)=-20.
      YO(4)=0.
C KZM  IS MINIMUM STACKING ALTITUDE
      KZM=5000
CNAL IS AIRCRAFT PER ALTITUDE
      NAL=4
      KT=(ITC/NAL)-1
C ZA IS ASSIGNED STACKING ALTITUDE
      ZA=KT*1000+KZM
C IT IS AIRCRAFT IN UNFILLED CIRCLE
      IT=ITC-KT*NAL
      IF(IT.GT.NAL) IT=NAL
      XXA=XA-XO(IT)
      YYA=YA-YO(IT)
      ATD=ATAN2(XXA,YYA)
      DIST=SQRT(((XA-XO(IT))**2)+((YA-YO(IT))**2))
      ERR=DIST-RR(IT)
      IF(ERR.GT.1.) TD=-ATD
      IF(ERR.LT.1.)TD=ATD+PI
      ERR=-ERR
      IF(ERR.GT.1.) TD=ATD
      RETURN
      END
$IBFTC THREE
      SUBROUTINE EXPECT(XA,YA,ZA,TD,VD,PD)
C THIS SUBROUTINE UPDATES POSITION
      COMMON/U/DT,TI
      COMMON/A/ISENSE
      COMMON/B/COUNT
      DIMENSION COUNT(10)
      DIMENSION ISENSE(4)
C VH HORIZONTAL VELOCITY
      VH=VD*COS(PD)
```

```
      XA=XA+(VH*DT*COS(TD))
      YA=YA+(VH*DT*SIN(TD))
      ZA=ZA/5280.
      ZA=ZA-(VD*DT*SIN(PD))
      ZA=ZA*5280.
      RETURN
      END
$IBFTC FOUR
      SUBROUTINE PLANE(X,Y,Z,T,V,P)
C THIS SUBROUTINE UPDATES AIRCRAFT POSITION
      COMMON/A/ISENSE
      COMMON/B/COUNT
      DIMENSION COUNT(10)
      DIMENSION ISENSE(4)
C SUBROUTINE ERROR INPUTS WIND AND OTHER ERRORS TO VELOCITY AND HEADING
      CALL ERROR(V,P,T)
C SUBROUTINE EXPECT GENERATES NEW POSITION
      CALL EXPECT(X,Y,Z,T,V,P)
      RETURN
      END
$IBFTC FIVE
      SUBROUTINE ERROR(V,R,T)
C THIS SUBROUTINE INPUTS ERROR
      COMMON/A/ISENSE
      COMMON/B/COUNT
      DIMENSION COUNT(10)
      DIMENSION ISENSE(4)
C VE  MAXIMUM VELOCITY ERROR
      VE =0.1
C TE MAXIMUM BEARING ERROR
      TE=0.01
C RANDOM GENERATING FUNCTION FOR VELOCITY ERROR. ANOTHER DISTRIBUTION MAY BE
C SUBSTITUTED FOR THIS OR ANY OTHER GENERATING FUNCTION.
      P=RAN(JX)
      S=RAN(JX)
      IF(S.GT.0.5) P=-P
      V=V+VE*P
C RANDOMLY GENERATED BEARING ERROR.
      P=RAN(JX)
      S=RAN(JX)
      IF(S.GT.0.5) P=-P
      T=T+TE*P
C RANDOMLY GENERATED CLIMB ERROR.
      P=RAN(JX)
      S=RAN(JX)
      IF(S.GT.0.5) P=-P
      R=R+TE*P
      RETURN
      END
$IBFTC SIX
      SUBROUTINE RUNWAY(J,XA,YA,ZA,TD,VD,PD)
C THIS SUBROUTINE SETS RUWAY CRITERIA COMMON TO ALL RUNWAYS.
       COMMON/R/IDO
      COMMON/P/ XS,YS,ZS,T
      COMMON /M/MTRAN
      COMMON/A/ISENSE
      COMMON/B/COUNT
      DIMENSION IDO(4)
      DIMENSION XS(4),YS(4),ZS(4),T(4)
```

```
      DIMENSION MTRAN(144)
      DIMENSION COUNT(10)
      DIMENSION ISENSE(4)
C CALL SPECIFIC RUNWAY TO SET RUNWAY SPECIFIC CRITERIA.
      GO TO (1,2,3,4),J
    1        CALL RUNONE(PD,VD,XA,YA,ZA,TD)
      GO TO 10
    2        CALL RUNTWO(PD,VD,XA,YA,ZA,TD)
      GO TO 10
    3        CALL RUNTHR(PD,VD,XA,YA,ZA,TD)
      GO TO 10
    4        CALL RUNFOU(PD,VD,XA,YA,ZA,TD)
   10 IF(VD.LT.0.) RETURN
C VELOCITY CONTROL IN AIRPORT AREA MAY BE EXPANDED
      DISTB=4.0
      ALTM=900.
C VDAIR RESET FOR STOL AIRCRAFT
C
C STOL CORRECTION
      VDAIR=280.
      IF(J.GT.2) VDAIR=140.
      DISTO=3.0
C INTRA RUNWAY AIRCRAFT SPACING
      IF(J.NE.JO) GO TO 20
      ALT=ABS(ZA-ZAO)
      ZAO=ZA
      IF(ALT.GT.ALTM) GO TO 20
      DIST=SQRT(((XA-XAO)**2)+((YA-YAO)**2))
      DTA=-1.
      TEMP= VD*COS(PD)*COS(TD)
      IF(XAO.GT.XA.AND.TEMP.GT.0.) DTA=1.
      IF(XAO.LT.XA.AND.TEMP.LT.0.) DTA=1.
      TEMP= VD*COS(PD)*SIN(TD)
      DTY=-1.
      IF(YAO.GT.YA.AND.TEMP.GT.0.) DTY=-1.
      IF(YAO.LT.YA.AND.TEMP.LT.0.) DTY=-1.
      DTA=DTA/DTY
      XAO=XA
      YAO=YA
      VD=60.*VD
      IF(DIST.LT.DISTO.AND.DTA.GE.0.) VD=VD-20.
      IF(DIST.LT.DISTO.AND.DTA.LE.0.) VD=VD+20.
      IF(DIST.GT.DISTB.AND.VD.LT.VDAIR) VD=VD+20.
      VDAIRX=VDAIR+40.
      IF(VD.GT.VDAIRX) VD =VD-20.
      IF(VD.LT.140.) VD=140.
      VD=VD/60.
   20 JO=J
      RETURN
      END
$IBFTC SEVEN
      SUBROUTINE RUNONE(PD,VD,XA,YA,ZA,TD)
C  THIS SUBROUTINE SETS CRITERIA FOR RUNWAY ONE.
      COMMON /R/ IDO
      COMMON/A/ISENSE
      COMMON/B/COUNT
      COMMON/P/ XS,YS,ZS,T
      COMMON /M/MTRAN
      DIMENSION XS(4),YS(4),ZS(4),T(4)
```

```
      DIMENSION IDO(4)
      DIMENSION MTRAN(144)
      DIMENSION ISENSE(4)
      DIMENSION COUNT(10)
  102 FORMAT (6F10.5)
      JRUN=IDO(1)
      TDU=TD
      IF(ISENSE(1).LT.1) GO TO 103
      CALL INPUT(XRS,YRS,THETA,THETB,DAP,DDI,ERROR,JRUN,RLEN,WAKE)
      ISENSE(1)=0
C INPUT RUNWAY GEOMETRY AND ACCEPTABLE ERROR
C XAM,YAM IS THE APPROACH OUTTER MARKER
C DAP IS APPROACH DISTANCE
C RLEN IS RUNWAY LENGTH
C DDI IS THE DEPARTURE OUTTER MARKER DISTANCE
C THETA ANGLE OF RUNWAY WITH NORTH
C ERROR ACCEPTABLE DEVIATION IN APPROACH.
C ST. LOUIS LAMBERT FIELD RUNWAY 12R
C     WRITE(6,102)XRS,YRS,DAP,RLEN,DDI,THETA,ERROR
      YMIN=YRS
      YMAX=YRS
      XMIN=XRS
      XMAX=XRS
      XAM=XRS +DAP*SIN(THETB)
      YAM=YRS +DAP*COS(THETB)
      IF(XAM.GT.XMAX) XMAX=XAM
      IF(XAM.LT. XMIN) XMIN=XAM
      IF(YAM.GT.YMAX) YMAX=YAM
      IF(YAM.LT. YMIN) YMIN=YAM
      XRE=XAM-(RLEN+DAP)*SIN(THETA)
      YRE=YAM-YRLEN+DAP)*COS(THETA)
      IF(XRE.LT.XMIN) XMIN=XRE
      IF(XRE.GT.XMAX) XMAX=XRE
      IF(YRE.LT.YMIN) YMIN=YRE
      IF(YRE.GT.YMAX) YMAX=YRE
      XDM=XAM-(RELEN+DAP+DDI)*SIN(THETA)
      YDM=YAM-(RELEN+DAP+DDI)*COS(THETA)
      IF(XDM.GT.XMAX) XMAX=XDM
      IF(XDM.LT.XMIN) XMIN=XDM
      IF(YDM.LT.YMIN) YMIN=YDM
      IF(YMM.GT.YMAX) YMAX=YDM
      YAB=(YRS+YRE)/2.
      IYA=YRS*1500./40.
      IXA=XRS*1500./40.
      JXA=XRE*1500./40.
      JYA=YRE*1500./40.
      MTRAN(2)=IXA
      MTRAN(3)=IYA
      MTRAN(4)=JXA
      MTRAN(5)=JYA
  103 PI=3.14159
C JRUN NEGATIVE IS DEPARTURE CODE
      IF(JRUN.GT.0 ) GO TO 7
C DEPARTURE PATTERN
        T(1)=THETA+PI
      XS(1)=XRE
      YS(1)=YRE
      ZS(1)=0.
      DP=SQRT(((XA-XRE)**2)+((YA-YRE)**2))
```

```
      CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(YA.GT.YMAX) TD=PI/2.
      IF(YA.LT.YMIN) TD=-PI/2.
      XREL=XA/XAM
      IF(XREL.GE.1.) TD=-PI/2.
      IF(XREL.LT.1.) TD=PI/2.
      IF(ZA.LT.800.) TD=THETA+PI
      VD=60.*VD
      IF(ZA.LT.800..AND.VD.LT.140.) VD=VD+20.
      IF(ZA.LT.800.) VD=-VD
      VD=VD/60.
             CALL SID(XA,YA,ZA,TD,PD,YDM,XDM,JRUN)
      RETURN
C VH HORIZONTAL VELOCITY
    7 VH=VD*COS(PD)
C APPROACH PATTERN
C DISM IS MINIMUM TURN DISTANCE
      DISM=10.
C DIST IS DISTANCE TO APPROACH MARKER
      DIST=SQRT(((XA-XAM)**2)+((YA-YAM)**2))
      IF(DIST.LT.DISM) GO TO 20
      YCA=ABS(YA-YAM)
      XCA=ABS(XA-XAM)
      IF(YCA.LT.DISM.OR.XCA.LT.DISM) GO TO 20
C COMPUTE DESIRED VECTOR
      IF(YA.GT.YMAX.AND.XA.GT.XMAX)TD=-PI/2.
      IF(YA.GT.YMAX.AND.XA.LT.XMIN)TD=-PI/2.
      IF(YA.LT.YMAX.AND.XA.LT.XMIN)TD= PI/2.
      IF(YA.LT.YMAX.AND.XA.GT.XMAX)TD= PI/2.
      YCA=ABS(YA-YMAX)
      IF(YCA.LT..05.AND.XA.GT.XMAX) TD=PI
      IF(YCA.LT..05.AND.XA.LT.XMIN) TD=0.
      IF(XA.LT.XMIN.OR.XA.GT.XMAX) RETURN
      IF(ANGLE.GT.0.) TD=0.
      IF(ANGLE.LT.0.) TD=PI
      IF(YA.LT.YMIN.OR.YA.GT.YMAX) RETURN
C COMPUTE ANGLE TO OUTTER MARKER
   20 SINTD=(YA-YAM)/DIST
      ATD=ARSIN(SINTD)
      IF(ATD.GT.0..AND.XA.LT.XAM)ATD =ATD+PI/2.
      IF(ATD.LT.0..AND.XA.LT.XAM)ATD =ATD-PI/2.
      PII=-PI/2.
      IF(THETA.GT.0..AND.ATD.GT.PII) TD=ATD+PI
      IF(THETA.GT.0..AND.ATD.GT.PII) RETURN
      IF(ATD.GT.0..AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.GT.0..AND.THETA.LT.0.) RETURN
      IF(ATD.LT.PII.AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.LT.PII.AND.THETA.LT.0.) RETURN
C IF ABORTING SET DECENT RATE TO ZERO
      PD=0.
      IF(YA.LT.YAB.AND.THETA.LT.0.) TD=-PI/2.
      IF(YA.LT.YAB.AND.THETA.GT.0.) TD= PI/2.
      IF(YA.LT.YAB) RETURN
C ENTERING APPROACH PATH TURN
C CORRECT APPROACH ANGLE TO RUNWAY END
      DP=SQRT(((XA-XRS)**2)+((YA-YRS)**2))
C CHECK FOR CONFLICT WITH DEPENDENT RUNWAY
             CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(DP.EQ.0.) DP=1.E-05
```

```
      SINTD=(YA-YRS)/DP
      TD=ARSIN(SINTD)
      IF(XA.LT.XRS) TD=TD+PI/2.
      TD=TD+PI
C COMPUTE DECENT RATE
      ZA=ZA/5280.
      TANPD=ZA/DP
      ZA=5280.*ZA
      PD=ATAN(TANPD)
      PDMAX=3.*PI/180.
      IF(PD.GT.PDMAX) PD =PDMAX
      VD=60.*VD
      IF(VD.GT.130.) VD=VD-20.
      VD=-VD/60.
      RETURN
      END
$IBFTC EIGHT
      SUBROUTINE RUNTWO(PD,VD,XA,YA,ZA,TD)
C   THIS SUBROUTINE SETS CRITERIA FOR RUNWAY TWO
      COMMON /R/ IDO
      COMMON/A/ISENSE
      COMMON/B/COUNT
      COMMON/P/ XS,YS,ZS,T
      COMMON /M/MTRAN
      DIMENSION XS(4),YS(4),ZS(4),T(4)
      DIMENSION IDO(4)
      DIMENSION MTRAN(144)
      DIMENSION ISENSE(4)
      DIMENSION COUNT(10)
  102 FORMAT (6F10.5)
      TDU=TD
      IF(ISENSE(2).LT.1) GO TO 103
      JRUN=2*IDO(2)
            CALL INPUT(XRS,YRS,THETA,THETB,DAP,DDI,ERROR,JRUN,RLEN,WAKE)
      ISENSE(2)=0
C INPUT RUNWAY GEOMETRY AND ACCEPTABLE ERROR
C XAM,YAM IS THE APPROACH OUTTER MARKER
C DAP IS APPROACH DISTANCE
C RLEN IS RUNWAY LENGTH
C DDI IS THE DEPARTURE OUTTER MARKER DISTANCE
C THETA ANGLE OF RUNWAY WITH NORTH
C ERROR ACCEPTABLE DEVIATION IN APPROACH.
C ST. LOUIS LAMBERT FIELD RUNWAY 12R
C ST. LOUIS LAMBERT FIELD RUNWAY 12L
C     WRITE(6,102)XRS,YRS,DAP,RLEN,DDI,THETA,ERROR
      YMIN=YRS
      YMAX=YRS
      XMIN=XRS
      XMAX=XRS
      XAM=XRS +DAP*SIN(THETB)
      YAM=YRS +DAP*COS(THETB)
      IF(XAM.GT.XMAX) XMAX=XAM
      IF(XAM.LT. XMIN) XMIN=XAM
      IF(YAM.GT.YMAX) YMAX=YAM
      IF(YAM.LT. YMIN) YMIN=YAM
      XRE=XAM-(RLEN+DAP)*SIN(THETA)
      YRE=YAM-(RLEN+DAP)*COS(THETA)
      IF(XRE.LT.XMIN) XMIN=XRE
      IF(XRE.GT.XMAX) XMAX=XRE
```

```
      IF(YRE.LT.YMIN) YMIN=YRE
      IF(YRE.GT.YMAX) YMAX=YRE
      XDM=XAM-(RELEN+DAP+DDI)*SIN(THETA)
      YDM=YAM-(RELEN+DAP+DDI)*COS(THETA)
      IF(XDM.GT.XMAX) XMAX=XDM
      IF(XDM.LT.XMIN) XMIN=XDM
      IF(YDM.LT.YMIN) YMIN=YDM
      IF(YMM.GT.YMAX) YMAX=YDM
      YAB=(YRS+YRE)/2.
      IXA=XRS*1500./40.
      IYA=YRS*1500./40.
      JXA=XRE*1500./40.
      JYA=YRE*1500./40.
      MTRAN(6)=IXA
      MTRAN(7)=IYA
      MTRAN(8)=JXA
      MTRAN(9)=JYA
  103 PI=3.14159
C JRUN NEGATIVE IS DEPARTURE CODE
      IF(JRUN.GT.0 ) GO TO 7
C DEPARTURE PATTERN
       T(2)=THETA+PI
      XS(2)=XRE
      YS(2)=YRE
      ZS(2)=0.
      DP=SQRT(((XA-XRE)**2)+((YA-YRE)**2))
      CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(YA.GT.YMAX) TD=PI/2.
      IF(YA.LT.YMIN) TD=-PI/2.
      XREL=XA/XAM
      IF(XREL.GE.1.) TD=-PI/2.
      IF(XREL.LT.1.) TD=PI/2.
      IF(ZA.LT.800.) TD=THETA+PI
      VD=60.*VD
      IF(ZA.LT.800..AND.VD.LT.140.) VD=VD+20.
      IF(ZA.LT.800.) VD=-VD
      VD=VD/60.
             CALL SID(XA,YA,ZA,TD,PD,YDM,XDM,JRUN)
      RETURN
C VH HORIZONTAL VELOCITY
    7 VH=VD*COS(PD)
C APPROACH PATTERN
C DISM IS MINIMUM TURN DISTANCE
      DISM=10.
C DIST IS DISTANCE TO APPROACH MARKER
      DIST=SQRT(((XA-XAM)**2)+((YA-YAM)**2))
      IF(DIST.LT.DISM) GO TO 20
      YCA=ABS(YA-YAM)
      XCA=ABS(XA-XAM)
      IF(YCA.LT.DISM.OR.XCA.LT.DISM) GO TO 20
C COMPUTE DESIRED VECTOR
      IF(YA.GT.YMAX.AND.XA.GT.XMAX)TD=-PI/2.
      IF(YA.GT.YMAX.AND.XA.LT.XMIN)TD=-PI/2.
      IF(YA.LT.YMAX.AND.XA.LT.XMIN)TD= PI/2.
      IF(YA.LT.YMAX.AND.XA.GT.XMAX)TD= PI/2.
      YCA=ABS(YA-YMAX)
      IF(YCA.LT..05.AND.XA.GT.XMAX) TD=PI
      IF(YCA.LT..05.AND.XA.LT.XMIN) TD=0.
      IF(XA.LT.XMIN.OR.XA.GT.XMAX) RETURN
```

```
      IF(ANGLE.GT.0.) TD=0.
      IF(ANGLE.LT.0.) TD=PI
      IF(YA.LT.YMIN.OR.YA.GT.YMAX) RETURN
C COMPUTE ANGLE TO OUTTER MARKER
   20 SINTD=(YA-YAM)/DIST
      ATD=ARSIN(SINTD)
      IF(ATD.GT.0..AND.XA.LT.XAM)ATD =ATD+PI/2.
      IF(ATD.LT.0..AND.XA.LT.XAM)ATD =ATD-PI/2.
      PII=-PI/2.
      IF(THETA.GT.0..AND.ATD.GT.PII) TD=ATD+PI
      IF(THETA.GT.0..AND.ATD.GT.PII) RETURN
      IF(ATD.GT.0..AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.GT.0..AND.THETA.LT.0.) RETURN
      IF(ATD.LT.PII.AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.LT.PII.AND.THETA.LT.0.) RETURN
C IF ABORTING SET DECENT RATE TO ZERO
      PD=0.
      IF(YA.LT.YAB.AND.THETA.GT.0.) TD= PI/2.
      IF(YA.LT.YAB.AND.THETA.LT.0.) TD=-PI/2.
      IF(YA.LT.YAB) RETURN
C ENTERING APPROACH PATH TURN
C CORRECT APPROACH ANGLE TO RUNWAY END
      DP=SQRT(((XA-XRS)**2)+((YA-YRS)**2))
C CHECK FOR CONFLICT WITH DEPENDENT RUNWAY
              CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(DP.EQ.0.) DP=1.E-05
      SINTD=(YA-YRS)/DP
      TD=ARSIN(SINTD)
      IF(XA.LT.XRS) TD=TD+PI/2.
      TD=TD+PI
C COMPUTE DECENT RATE
      ZA=ZA/5280.
      TANPD=ZA/DP
      ZA=5280.*ZA
      PD=ATAN(TANPD)
      PDMAX=3.*PI/180.
      IF(PD.GT.PDMAX) PD =PDMAX
      VD=60.*VD
      IF(VD.GT.130.) VD=VD-20.
      VD=-VD/60.
      RETURN
      END
$IBFTC NINE
      SUBROUTINE RUNTHR(PD,VD,XA,YA,ZA,TD)
C  THIS SUBROUTINE SETS CRITERIA FOR RUNWAY THREE
      COMMON /R/ IDO
      COMMON /M/MTRAN
      COMMON/A/ISENSE
      COMMON/B/COUNT
      COMMON/P/ XS,YS,ZS,T
      DIMENSION MTRAN(144)
      DIMENSION IDO(4)
      DIMENSION XS(4),YS(4),ZS(4),T(4)
      DIMENSION ISENSE(4)
      DIMENSION COUNT(10)
  102 FORMAT (6F10.5)
      TDU=TD
      IF(ISENSE(3).LT.1) GO TO 103
      JRUN=3*IDO(3)
```

```
              CALL INPUT(XRS,YRS.THETA,THETB,DAP,DDI,ERROR,JRUN,RLEN,WAKE)
         ISENSE(3)=0
C INPUT RUNWAY GEOMETRY AND ACCEPTABLE ERROR
C XAM,YAM IS THE APPROACH OUTTER MARKER
C DAP IS APPROACH DISTANCE
C RLEN IS RUNWAY LENGTH
C DDI IS THE DEPARTURE OUTTER MARKER DISTANCE
C THETA ANGLE OF RUNWAY WITH NORTH
C ERROR ACCEPTABLE DEVIATION IN APPROACH.
C ST. LOUIS LAMBERT FIELD RUNWAY STOL LAND
C STOL RUNWAY MUST BE A MINIMUM OF 2500. FEET FROM OTHER
C RUNWAYS DUE TO WAKE TURBULENCE
      XRS=0.0-WAKE*COS(THETA)
      YRS=YRS+WAKE*SIN(THETA)
      DCR=8000./5280.
      XRS=XRS-DCR*SIN(THETA)
      YRS=YRS-DCR*COS(THETA)
C     WRITE(6,102)XRS,YRS,DAP,RLEN,DDI,THETA,ERROR
      YMIN=YRS
      YMAX=YRS
      XMIN=XRS
      XMAX=XRS
      XAM=XRS +DAP*SIN(THETB)
      YAM=YRS +DAP*COS(THETB)
      IF(XAM.GT.XMAX) XMAX=XAM
      IF(XAM.LT.XMIN) XMIN=XAM
      IF(YAM.GT.YMAX) YMAX=YAM
      IF(YAM.LT.YMIN) YMIN=YAM
      XRE=XRS -(RLEN)*SIN(THETA)
      YRE=YRS -(RLEN)*COS(THETA)
      IF(XRE.LT.XMIN) XMIN=XRE
      IF(XRE.GT.XMAX) XMAX=XRE
      IF(YRE.LT.YMIN) YMIN=YRE
      IF(YRE.GT.YMAX) YMAX=YRE
      XDM=XRS-(RLEN+DDI)*SIN(THETA)
      YDM=YRS-(RLEN+DDI)*COS(THETA)
      IF(XDM.GT.XMAX) XMAX=XDM
      IF(XDM.LT.XMIN) XMIN=XDM
      IF(YDM.LT.YMIN) YMIN=YDM
      IF(YMM.GT.YMAX) YMAX=YDM
      YAB=(YRS+YRE)/2.
      IXA=XRS*1500./40.
      IYA=YRS*1500./40.
      JXA=XRE*1500./40.
      JYA=YRE*1500./40.
      MTRAN(14)=IXA
      MTRAN(15)=IYA
      MTRAN(16)=JXA
      MTRAN(17)=JYA
  103 PI=3.14159
C JRUN NEGATIVE IS DEPARTURE CODE
      IF(JRUN.GT.0 ) GO TO 7
C DEPARTURE PATTERN
      ZS(3)=0.
      YS(3)=YRE
      XS(3)=XRE
       T(3)=THETA+PI
      DP=SQRT(((XA-XRE)**2)+((YA-YRE)**2))
      CALL INTCON(XAM,YAM,JRUN,DP,IAB)
```

```
      IF(YA.GT.YMAX) TD=PI/2.
      IF(YA.LT.YMIN) TD=-PI/2.
      XREL=XA/XAM
      IF(XREL.GE.1.) TD=-PI/2.
      IF(XREL.LT.1.) TD=PI/2.
      IF(ZA.LT.800.) TD=THETA+PI
      VD=60.*VD
      IF(ZA.LT.800..AND.VD.LT.140.) VD=VD+20.
      IF(ZA.LT.800.) VD=-VD
      VD=VD/60.
            CALL SID(XA,YA,ZA,TD,PD,YDM,XDM,JRUN)
      RETURN
C VH HORIZONTAL VELOCITY
    7 VH=VD*COS(PD)
C APPROACH PATTERN
C DISM IS MINIMUM TURN DISTANCE
      DISM=10.
C DIST IS DISTANCE TO APPROACH MARKER
      DIST=SQRT(((XA-XAM)**2)+((YA-YAM)**2))
      IF(DIST.LT.DISM) GO TO 20
      YCA=ABS(YA-YAM)
      XCA=ABS(XA-XAM)
      IF(YCA.LT.DISM.OR.XCA.LT.DISM) GO TO 20
C COMPUTE DESIRED VECTOR
      IF(YA.GT.YMAX.AND.XA.GT.XMAX)TD=-PI/2.
      IF(YA.GT.YMAX.AND.XA.LT.XMIN)TD=-PI/2.
      IF(YA.LT.YMAX.AND.XA.LT.XMIN)TD= PI/2.
      IF(YA.LT.YMAX.AND.XA.GT.XMAX)TD= PI/2.
      YCA=ABS(YA-YMAX)
      IF(YCA.LT..05.AND.XA.GT.XMAX) TD=PI
      IF(YCA.LT..05.AND.XA.LT.XMIN) TD=0.
      IF(XA.LT.XMIN.OR.XA.GT.XMAX) RETURN
      IF(ANGLE.GT.0.) TD=0.
      IF(ANGLE.LT.0.) TD=PI
      IF(YA.LT.YMIN.OR.YA.GT.YMAX) RETURN
C COMPUTE ANGLE TO OUTTER MARKER
   20 SINTD=(YA-YAM)/DIST
      ATD=ARSIN(SINTD)
      IF(ATD.GT.0..AND.XA.LT.XAM)ATD =ATD+PI/2.
      IF(ATD.LT.0..AND.XA.LT.XAM)ATD =ATD-PI/2.
      PII=-PI/2.
      IF(THETA.GT.0..AND.ATD.GT.PII) TD=ATD+PI
      IF(THETA.GT.0..AND.ATD.GT.PII) RETURN
      IF(ATD.GT.0..AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.GT.0..AND.THETA.LT.0.) RETURN
      IF(ATD.LT.PII.AND.THETA.LT.0.) TD=ATD+PI
      IF(ATD.LT.PII.AND.THETA.LT.0.) RETURN
C IF ABORTING SET DECENT RATE TO ZERO
      PD=0.
      IF(YA.LT.YAB.AND.THETA.LT.0.) TD=-PI/2.
      IF(YA.LT.YAB.AND.THETA.GT.0.) TD= PI/2.
      IF(YA.LT.YAB) RETURN
C ENTERING APPROACH PATH TURN
C CORRECT APPROACH ANGLE TO RUNWAY END
      DP=SQRT(((XA-XRS)**2)+((YA-YRS)**2))
C CHECK FOR CONFLICT WITH DEPENDENT RUNWAY
            CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(DP.EQ.0.) DP=1.E-05
      SINTD=(YA-YRS)/DP
```

```
      TD=ARSIN(SINTD)
      IF(XA.LT.XRS) TD=TD+PI/2.
      TD=TD+PI
C COMPUTE DECENT RATE
      ZA=ZA/5280.
      TANPD=ZA/DP
      ZA=5280.*ZA
      PD=ATAN(TANPD)
      PDMAX=8.*PI/180.
      IF(PD.GT.PDMAX) PD =PDMAX
C RUNWAY APPROACH VELOCITY CONTROL
      VD=60.*VD
      IF(VD.GT. 65.) VD=VD-20.
      VD=-VD/60.
      RETURN
      END
$IBFTC TEN
      SUBROUTINE RUNFOU(PD,VD,XA,YA,ZA,TD)
C  THIS SUBROUTINE SETS CRITERIA FOR RUNWAY FOUR.
      COMMON /P/ IDO
      COMMON /M/MTRAN
      COMMON/A/ISENSE
      COMMON/B/COUNT
      COMMON/P/ XS,YS,ZS,T
      DIMENSION MTRAN(144)
      DIMENSION XS(4),YS(4),ZS(4),T(4)
      DIMENSION ISENSE(4)
      DIMENSION COUNT(10)
      DIMENSION IDO(4)
  102 FORMAT (6F10.5)
      TDU=TD
      IF(ISENSE(4).LT.1) GO TO 103
      JRUN=4*IDO(4)
              CALL INPUT(XRS,YRS,THETA,THETB,DAP,DDI,ERROR,JRUN,RLEN,WAKE)
      ISENSE(4)=0
C INPUT RUNWAY GEOMETRY AND ACCEPTABLE ERROR
C XAM,YAM IS THE APPROACH OUTTER MARKER
C DAP IS APPROACH DISTANCE
C RLEN IS RUNWAY LENGTH
C DDI IS THE DEPARTURE OUTTER MARKER DISTANCE
C THETA ANGLE OF RUNWAY WITH NORTH
C ERROR ACCEPTABLE DEVIATION IN APPROACH.
C ST. LOUIS LAMBERT FIELD RUNWAY STOL DEPAT
C STOL RUNWAY MUST BE A MINIMUM OF 2500. FEET FROM OTHER
C RUNWAYS DUE TO WAKE TURBULENCE
      XRS=0.0-WAKE*COS(THETA)
      YRS=YRS+WAKE*SIN(THETA)
      DCR=8000./5280.
      XRS=XRS-DCR*SIN(THETA)
      YRS=YRS-DCR*COS(THETA)
C     WRITE(6,102)XRS,YRS,DAP,RLEN,DDI,THETA,ERROR
      YMIN=YRS
      YMAX=YRS
      XMIN=XRS
      XMAX=XRS
      XAM=XRS +DAP*SIN(THETB)
      YAM=YRS +DAP*COS(THETB)
      IF(XAM.GT.XMAX) XMAX=XAM
      IF(XAM.LT.XMIN) XMIN=XAM
```

```
      IF(YAM.GT.YMAX) YMAX=YAM
      IF(YAM.LT.YMIN) YMIN=YAM
      YRE=YRS -(RLEN)*COS(THETA)
      XRE=XRS -(RLEN)*SIN(THETA)
      IF(XRE.LT.XMIN) XMIN=XRE
      IF(XRE.GT.XMAX) XMAX=XRE
      IF(YRE.LT.YMIN) YMIN=YRE
      IF(YRE.GT.YMAX) YMAX=YRE
      XDM=XRS-(RLEN+DDI)*SIN(THETA)
      YDM=YRS-(RLEN+DDI)*COS(THETA)
      IF(XDM.GT.XMAX) XMAX=XDM
      IF(XDM.LT.XMIN) XMIN=XDM
      IF(YDM.LT.YMIN) YMIN=YDM
      IF(YMM.GT.YMAX) YMAX=YDM
      YAB=(YRS+YRE)/2.
      IXA=XRS*1500./40.
      IYA=YRS*1500./40.
      JXA=XRE*1500./40.
      JYA=YRE*1500./40.
      MTRAN(10)=IXA
      MTRAN(11)=IYA
      MTRAN(12)=JXA
      MTRAN(13)=JYA
  103 PI=3.14159
C JRUN NEGATIVE IS DEPARTURE CODE
      IF(JRUN.GT.0 ) GO TO 7
C DEPARTURE PATTERN
      ZS(4)=0.
      YS(4)=YRE
      XS(4)=XRE
       T(4)=THETA+PI
      DP=SQRT(((XA-XRE)**2)+((YA-YRE)**2))
      CALL INTCON(XAM,YAM,JRUN,DP,IAB)
      IF(YA.GT.YMAX) TD=PI/2.
      IF(YA.LT.YMIN) TD=-PI/2.
      XREL=XA/XAM
      IF(XREL.GE.1.) TD=-PI/2.
      IF(XREL.LT.1.) TD=PI/2.
      IF(ZA.LT.800.) TD=THETA+PI
      VD=60.*VD
      IF(ZA.LT.800..AND.VD.LT.140.) VD=VD+20.
      IF(ZA.LT.800.) VD=-VD
      VD=VD/60.
              CALL SID(XA,YA,ZA,TD,PD,YDM,XDM,JRUN)
      RETURN
C VH HORIZONTAL VELOCITY
    7 VH=VD*COS(PD)
C APPROACH PATTERN
C DISM IS MINIMUM TURN DISTANCE
      DISM=10.
C DIST IS DISTANCE TO APPROACH MARKER
      DIST=SQRT(((XA-XAM)**2)+((YA-YAM)**2))
      IF(DIST.LT.DISM) GO TO 20
      YCA=ABS(YA-YAM)
      XCA=ABS(XA-XAM)
      IF(YCA.LT.DISM.OR.XCA.LT.DISM) GO TO 20
C COMPUTE DESIRED VECTOR
      IF(YA.GT.YMAX.AND.XA.GT.XMAX)TD=-PI/2.
      IF(YA.GT.YMAX.AND.XA.LT.XMIN)TD=-PI/2.
```

```
      COMMON/A/ISENSE
      COMMON/B/COUNT
      DIMENSION COUNT(10)    ,ISENSE(4)
      DIMENSION IDO(4),JDO(4),XXO(4),YYO(4) ,ICON(4,4)
      DCON=1.0
      DPM=2.0
      DPN=3.0
      J=IABS(JRUN)
      XXO(J)=XAO
      YYO(J)=YAO
      DO 7 I=1,4
      IF(ISENSE(I).GT.0) GO TO 7
      IF(I.EQ.J) GO TO 7
      DIST=SQRT(((XXO(J)-XXO(I))**2)+((YYO(J)-YYO(I))**2))
      IF(DIST.GT.DCON) GO TO 7
      ICON(J,I)=I
      ICON(I,J)=J
    7 CONTINUE
    5 DO 14 I=1,4
      K=ICON(J,I)
      IF(K.NE.I) GO TO 14
      IDO(K)=0
      IF(DP.LT.DPN.AND.JRUN.LT.0 ) IDO(K)=-1
      IF(DP.LT.DPM.AND.JRUN.GT.0) IDO(K)=-1
      IF(IDO(K).LT.0) ISENSE(I)=-4
   14 CONTINUE
      RETURN
      END
$IBFTC SIXT
      SUBROUTINE ARIVIT(J)
      COMMON /R/ IDO
      COMMON/P/ XS,YS,ZS,T
      DIMENSION XS(4),YS(4),ZS(4),T(4)
      DIMENSION IDO(4)
      IF(IDO(J).LT.0) RETURN
      PI=3.14159
      PIJ=-PI
      DISTBO=30.
      X=XS(J)
      Y=YS(J)
      D=SQRT((X**2)+(Y**2))
      IF(D.EQ.0.) D=1.E-05
      U=Y/D
      IF(U.GT.1..OR.U.LT.1.) U=1.
      U=ARSIN(U)
      U=U+(X+Y)/DISTBO
   10 IF(U.GT.PI) U=U-PI
      IF(U.LT.PIJ)U=U-PIJ
      IF(U.LT.PI.AND.U.GT.PIJ) GO TO 5
      GO TO 10
    5 X =DISTBO*SIN(U)
      Y =DISTBO*COS(U)
      XS(J)=X
      YS(J)=Y
       T(J)=U+PI
      RETURN
      END
$IBFTC SEVENT
      SUBROUTINE SID(XA,YA,ZA,TD,PD,YDM,XDM,JRUN)
```

```
      DIMENSION TU(6,6),DU(6),ZU(6,5)
      PI=3.14159
      DU(1)=2.0
      DU(2)=10.
      DU(3)=15.
      DU(4)=20.
      DU(5)=40.
      DO27 K=1,5
      TU(K,1)=PI
      DO 29 L=2,5
      TU(K,L)=TU(K,L-1) -PI/2.
   29 CONTINUE
   27 CONTINUE
      DO 51 K=1,5
      DO 52 L=1,5
      ZU(K,L)=1000.+1500.*(L-1)
   52 CONTINUE
   51 CONTINUE
      J=IABS(JRUN)
C ZMIN IS MINIMUM TURN ALTITUDE
      ZMIN=800.
C DMIN IS MINIMUM TURN DISTANCE FROM RUNWAY END
      DMIN=2.0
      DIST=(((XA-XDM)**2)+((YA-YDM)**2))
C IF MINIMUMS NOT MET RETURN
      IF(ZA.LT.ZMIN.OR.DIST.LT.DMIN) RETURN
      DIST=SQRT((XA**2)+(YA**2))
      DO 7 I=1,5
      JP=I+1
      IF(DIST.LT.DU(I)) GO TO 9
    7 CONTINUE
      RETURN
    9 IF (DIST.LE.0.) DIST=1.E-05
      TK=YA/DIST
      TV=ARSIN(TK)
      IF(TK.GT.0..AND.XA.LT.0.) TV=TV+PI
      IF(TK.LT.0..AND.XA.LT.0.) TV=TV-PI
      TD=TU(J,JR)
      PD=0.
      IF(ZA.LT.ZU(JR,J)) PD=-3.*PI/180.
      IF(J.GT.2) PD=(8./3.)*PD
      RETURN
      END
$IBFTC EIGHTT
      SUBROUTINE DUAL(TC,J)
      DIMENSION IC(4),TC(4)
      IC(1)=1
      IC(2)=2
      IC(3)=4
      IC(4)=3
      K=IC(J)
      IF(TC(J).GT.TC(K)) TC(K)=TC(J)
      IF(TC(K).GT.TC(J)) TC(J)=TC(K)
      RETURN
      END
$IBFTC NINET
      SUBROUTINE PLAND(  XA,YA,ZA,TD,VD,PD,J,IT,I)
C THIS SUBROUTINE GENERATES CONTROLLER EXPECTATION OF PLANE POSITION.
      COMMON/R/ IDO
```

```
      COMMON/A/ISENSE
      COMMON/B/COUNT
      COMMON/U/DT,TI
      COMMON/C/ZMAX,ZMIN,TMAX,TMIN
      COMMON/P/ XS,YS,ZS,T
      COMMON /M/MTRAN
      DIMENSION IDO(4)
      DIMENSION XS(4),YS(4),ZS(4),T(4)
      DIMENSION MTRAN(144)
      DIMENSION ICN(4)
      DIMENSION ZMAX(30),ZMIN(30),TMAX(30),TMIN(30)
      DIMENSION COUNT(10)
      DIMENSION ISENSE(4)
      DIMENSION ZAI(4)
      DIMENSION ZAA(4,30),TDA(4,30)
  102 FORMAT (6F10.5)
C ICN CONTROLS THE MAXIMUM NUMBER OF AIRCRAFT ASSIGNED TO RUNWAY APPROACH
      ICN(1)=5
      ICN(2)=5
      ICN(3)=5
      ICN(4)=5
C ZAI ALTITUDES ASSIGNED TO VARIOUS BEARINGS.
  103 ZAI(2)=2200.
      ZAI(4)=3200.
      ZAI(1)=2200.
      ZAI(3)=3200.
      PI=3.14159
C STORE BEARING AND ALTITUDE
      TDX=TD
      ZAX=ZA
C RUNWAY UPDATES CONTROLLER INSTRUCTIONS BASED ON POSITION DATA.
      IC=COUNT(10)
C NO ARRIVALS
      IU=IABS(IC)
C RESET STACKING COUNTER
      IF(IU.EQ.1.AND.J.EQ.1) ITC=1
      IF(IC.LT.0) GO TO 10
      IC=IABS(IC)
C IF EXCESS AIRCRAFT STACK THEM
      IF(IC.LT.ICN(J)) GO TO 10
      CALL STACK(ITC,XA,YA,ZA,TD)
      ITC=ITC+1
      GO TO 900
   10       CALL RUNWAY(J,XA,YA,ZA,TD,VD,PD)
C A NEGATIVE VD MEANS IN APPROACH CONTROLLER SECTOR
      IF(VD.LT.0.) GO TO 15
C EACH RUNWAY HAS A SPECIFIED ALTITUDE
      ZA=ZAI(J)
C COMPUTE DESIRED CLIMB.ZD
C  CONTROLLER WORK LOAD COUNT
  900 ERZ=ABS(ZA-ZAA(J,I))
      ETD=ABS(TD-TDA(J,I))
C IF A VECTOR CHANGE OCCURRED ADD TO THE COUNT(9)
      IF(ERZ.GT.100..OR.ETD.GT.0.15) COUNT(9)=COUNT(9)+1.
      ZAA(J,I)=ZA
      TDA(J,I)=TD
      ZD=(ZA-ZAX)/5280.
C COMPUTE MAXIMUM CLIMB ZP        MAXIMUM DECENT ZC
      ZP=ZMAX(IT)*DT/5280.
```

```
      ZC=ZMIN(IT)*DT/5280.
      VDA=VD*DT
      IF(VDA.EQ.0.) VDA=1.E-05
      IF(ZP.GT.VDA) ZP=VDA
      VDA=-VDA
      IF(ZC.LT.VDA) ZC=VDA
C COMPUTE DESIRED CLIMB ANGLE
C IF CLIMB EXCEEDS AIRCRAFT CAPABILITY DECREASE EXPECTATIONS.
      IF(ZD.GT.ZP)ZD=ZP
      IF(ZD.LT.ZC) ZD=ZC
      SINPD=ZD/VDA
      PD=ARSIN(SINPD)
      PDMIN=-3.*PI/180.
      PDMAX= 3.*PI/180.
      IF(J.GT.2) PDMIN=-8.*PI/180.
      IF(J.GT.2) PDMAX=8.*PI/180.
      IF(PD.LT.PDMIN) PD=PDMIN
      IF(PD.GT.PDMAX) PD=PDMAX
C TC IS DESIRED TURN
      TC=TD-TDX
      IF(TC.GT.PI )TC=TC-2.*PI
      PIX=-PI
      IF(TC.LT.PIX) TC=2.*PI+TC
C IF DESIRED TURN EXCEEDS AIRCRAFT CAPABILITY REDUCE EXPECTATIONS
      TXX=DT*TMAX(IT)
      TNN=DT*TMIN(IT)
      IF(TC.GT.TXX )TC=TXX
      IF(TC.LT.TNN ) TC=TNN
C UPDATE EXPECTED BEARING.
      TD=TDX+TC
C UPDATE EXPECTED POSITION.
   15 VD=ABS(VD)
             CALL EXPECT(XA,YA,ZA,TD,VD,PD)
      RETURN
      END
$IBFTC TWENTY
      SUBROUTINE DATA(N)
      COMMON /LU/ WAK
      COMMON/LU/XA,YA,TTA,TTB,DA,DD,ER,RL
      COMMON /R/ IDO
      DIMENSION IDO(4)
      DIMENSION XA(4),YA(4),TTA(4),TTB(4),DA(4),DD(4),ER(4),RL(4)
      READ(5,29)
C     WRITE(6,29)
   29 FORMAT(1X,'TEST RUN')
      READ(5,10) N ,WAK
      WAKE=WAK
C     WRITE(6,12)N ,WAKE
      WAKE=WAKE/5280.
      IF(N.GT.4) N=4
   10 FORMAT(I10,F10.5)
   12 FORMAT(I10,F10.5)
   25 FORMAT(4I10)
      READ(5,25) (IDO(K),K=1,N)
      DO 20 I=1,N
      READ(5,11) XA(I),YA(I),TTA(I),TTB(I),DA(I),DD(I),ER(I) ,RL(I)
C     WRITE(6,11)XA(I),YA(I),TTA(I),TTB(I),DA(I),DD(I),ER(I) ,RL(I)
      XA(I)=XA(I)/5280.
      YA(I)=YA(I)/5280.
```

```
      RL(I)=RL(I)/5280.
   11 FORMAT(1X,F9.5,7F10.5)
   20 CONTINUE
      RETURN
$IBFTC TTWO
      SUBROUTINE INPUT(XRS,YRS,THETA,THETB,DAP,DDI,ERROR,K,RLEN,WAKE)
      COMMON/LU/XA,YA,TTA,TTB,DA,DD,ER,RL
      DIMENSION XA(4),YA(4),TTA(4),TTB(4),DA(4),DD(4),ER(4),RL(4)
      J=IABS(K)
      XRS=XA(J)
      YRS=YA(J)
      THETA =TTA(J)
      THETB=TTB(J)
      DAP=DA(J)
      DDI=DD(J)
      ERROR=ER(J)
      RLEN=RL(J)
      RETURN
      END
$IBFTC TONE
      SUBROUTINE CPXSN(IWORD,MTRAN,JUNN)
      DIMENSION MTRAN(144)
   12 FORMAT(10I10)
      WRITE(12,12) (MTRAN(I),I=1,144)
      RETURN
      END
$DATA
'     END OF FILE CARD
'     END OF FILE CARD
```