

UNCLASSIFIED

AD 409 025

DEFENSE DOCUMENTATION CENTER

FOR

SCIENTIFIC AND TECHNICAL INFORMATION

CAMERON STATION, ALEXANDRIA, VIRGINIA



UNCLASSIFIED

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

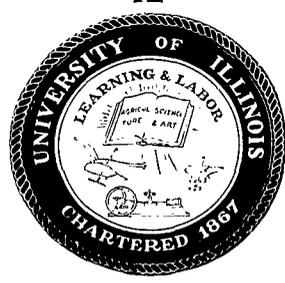
N-63-4-2

409 025

CATALOGED BY DDC
409025
AS AD No. _____

DDC
RECEIVED
JUL 18 1963
LIBRARY
T. A. A.

Coordinated
Science
Laboratory



UNIVERSITY OF ILLINOIS - URBANA, ILLINOIS

**AUTOMATIC AIR TRAFFIC CONTROL
PART II AN EXPERIMENTAL CONTROL
LOGIC**

L.S. Kypka & H. Bobotek

REPORT R-146

APRIL, 1963

COORDINATED SCIENCE LABORATORY
UNIVERSITY OF ILLINOIS
URBANA, ILLINOIS

Contract DA-36-039-TR US AMC 02208(E)
DA Project 3A-99-25-004

The research reported in this document was made possible by support extended to the University of Illinois, Coordinated Science Laboratory, jointly by the Department of the Army, Department of the Navy (Office of Naval Research), and the Department of the Air Force (Office of Scientific Research) under Department of Army Contract DA-36-039-TR US AMC 02208(E).

ABSTRACT

The work in automatic air traffic control conducted at CSL has consisted of three phases: 1) design of an automatic control system, 2) design of an experimental control logic, and 3) evaluation of the control logic in real-time simulation. The first phase has been described in Part I of this series of two reports. This report, i.e. Part II, describes the experimental control logic and the results of the simulation. The simulation was done on the Cornfield System which consists of a special purpose tracking computer (TASC) and a general purpose digital computer (ILLIAC) connected together. The main assumptions in the work are (a) perfect tracking, (b) zero winds aloft, and (c) all traffic operating at the same altitude. Simulated traffic consists of four types of aircraft ranging in cruising speed from 131 mph to 542 mph and the assumed method of steering is vectoring via digital data link. Cornfield System capacity of 25 aircraft is shown to have no relationship to the capacity of a system especially designed to execute automatic control.

Three control programs have been assembled to enable simulation of automatic and manual control in en route, approach, and terminal areas. The logic is such that en route traffic is steered directly toward destinations except for conflict resolution. In extreme cases, conflicts are resolved by resorting to a standard holding pattern. In approach area, all conflicts are resolved by holding and in terminal area, a unique method of passive collision avoidance is used.

ACKNOWLEDGEMENTS

The authors are indebted to Mrs. Patricia Hunt and Mr. Ernest Neff for their cooperation and assistance in the work described in this report. Mrs. Hunt did most of the programming and de-bugging and Mr. Neff maintained and operated the Cornfield System. Both also participated as controllers in the experiments in manual control. Further thanks are due Mr. J. Cummings and Mr. W. Philman for their patience and ingenuity in obtaining the photographs appearing in Appendix D. In addition, the authors wish to thank R. D. Jenks, R. M. Brown and D. H. Cooper for their constructive criticism and appraisal of this report.

TABLE OF CONTENTS

	<u>Page</u>
I. INTRODUCTION	7
II. EXPERIMENTAL FRAMEWORK	8
A. Cornfield System	10
1. ILLIAC	11
2. Tracking and Sorting Computer.	12
3. Display.	12
4. ILLIAC Input	13
5. Manual Input to ILLIAC	13
6. Target Insertion	13
7. Limitations.	14
B. Assumptions and Restrictions	15
1. Assumptions in Air Traffic	15
2. Assumptions in Environment (PICON)	17
3. Assumptions in Environment (PICON-ER).	18
4. Assumptions in Control System.	18
III. AUTOMATIC CONTROL	20
A. Basic Logical Functions.	21
1. Sort	22
2. Reference Aircraft Concept	22
3. Pairing Rule	23
4. Class Switching.	24
5. Variables.	25
6. ΔJ_1 Logic.	26
7. Collision Sorter	31
8. Conflict Detection	34
9. Conflict Resolution.	38
B. En route and Approach Control.	42
1. Shell.	43
2. Conflict Overrule.	47
3. En route Logic	48
4. Approach Control Logic	58
5. Uncontrolled Traffic (VFR)	63
6. Optimization Logic	64
C. Terminal Sequencing Control.	67
1. Assembling of the Landing Sequence	68
2. Regulation of Landing Separation	71
3. Collision Avoidance	79
4. Departures	84
5. Uncontrolled Traffic (VFR)	84

	<u>Page</u>
IV. EVALUATION AND CONCLUSIONS	86
A. En route Control	86
B. Approach and Terminal Control.	90
C. Computation Times.	92
D. Further Developments	94
V. APPENDIX A. DERIVATION OF VARIABLES.	97
VI. APPENDIX B. CONFLICT RESOLUTION SPECIAL CASE	98
VII. APPENDIX C. ESTIMATION OF LANDING TIME	102
VIII. APPENDIX D. PHOTOGRAPHS.	106

I. INTRODUCTION

The subject of this report is automatic decision making as applied to air traffic control. In this work, four phases of flight are recognized: departure, en route, approach, and terminal sequencing. It is our contention that all four phases of flight can be controlled by a system of logic residing in several special purpose digital machines on the ground. A design for the necessary digital machinery has been presented in Part I.

In addition to the system design referred to above, the work has extended to devising a control philosophy and studying it in operation in a partial, real-time simulation of the system described in Part I. The control philosophy assumes that all air space is available for use (no airways) and the technique of steering traffic is the same as presently-used radar vectoring (digital data link assumed).

The purpose of the work described here was primarily to isolate the problems of automatic control, solve them if possible, and in the process develop some good techniques for use in an automatic system. In terms of these goals, it is felt that considerable success has been found even though the logic itself leaves much to be desired. The reader is urged to note the distinction between basic logical functions (or tools) and the control logic. In an actual machine, the arithmetic unit would be wired to execute the basic logical functions and the control logic would be analogous to a program stored in the memory.

II. EXPERIMENTAL FRAMEWORK

Fig. 1 is a reproduction of the diagram of the projected system described in Part I of this report. It is reproduced here to help show exactly what is meant by "partial simulation" referred to in the introduction. Simulation of the entire system is well beyond the capabilities of the Cornfield System so it was necessary to resort to partial simulation.

It is noted from Fig. 1 that the common memory is connected to 4 sets of logic, each intended to perform its particular function independently of the others. Since each is independent, one or more of the sections can be eliminated from simulation as long as the effect on the common memory is not lost. In the work done at CSL, the manual input and display section was eliminated entirely. In its place the reader will find four assumptions: 1) all traffic is identified, 2) all traffic will respond to a heading instruction, 3) tracking is perfect, and 4) emergencies cannot occur. These assumptions eliminate the need for human intervention from a controller's console.

Internal communication logic is also eliminated if an experimenter assumes 1) only one automatic air traffic control center and control area exists and 2) the approach-departure computer is part of the en route control computer. These assumptions are accepted in this report and internal communication logic is not simulated.

Since the Cornfield System contains a tracking computer and a control computer, these two items are assumed to exist. Much is known about tracking logic and therefore no experimenting was done in this area; the tracking computer was simply used to simulate air traffic and in this capacity, it exhibits the perfect tracking property. Our partial simulation of the system of Fig. 1, therefore, consists of control logic, common memory, and tracking computer executing perfect tracking.

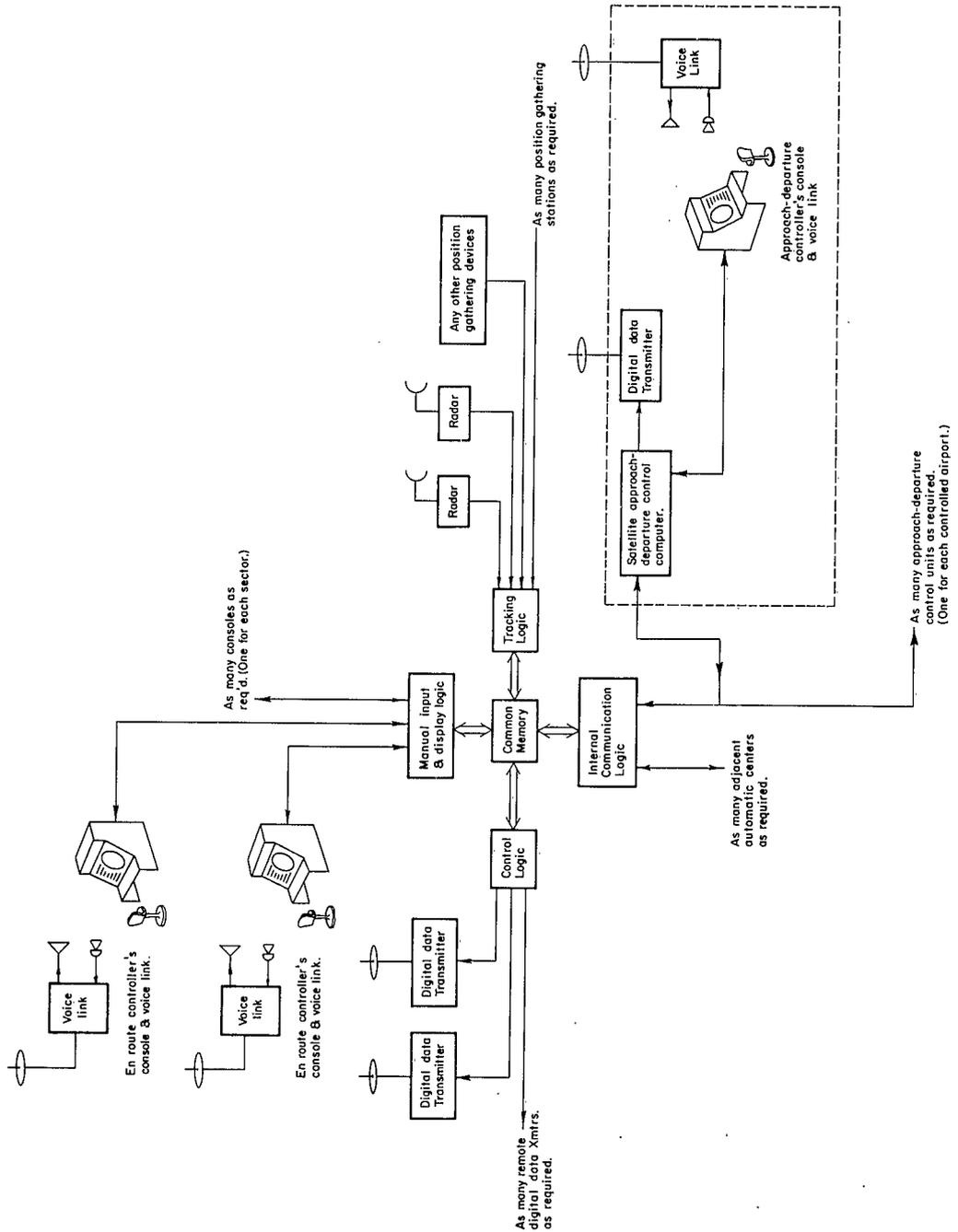


Fig. 1. Automatic air traffic control center showing one satellite computer.

A. Cornfield System

The Cornfield System was originally designed and built to be used in the study of automatic control of air traffic in a hostile military environment. The study was terminated in 1960 at which time the work described in this report was begun. As will be seen from the following description, the difference between "vectoring aircraft to close on hostile aircraft" and "vectoring to avoid closing" is simply a matter of writing programs for ILLIAC. The following description and diagram (Fig. 2) are brief but adequate for purposes of this report. Additional details of the Cornfield System can be found in the listed reports.

1. ILLIAC

ILLIAC is a general-purpose, medium-speed digital computer. Its order code is of the single, direct-address type with two complete orders per word. A word consists of 40 bits, and the active memory (Williams electrostatic) will store 1024 words. The accumulator consists of two 40 bit registers, one of which is also the input-output register. Input and output are executed serially by tetrads. The auxiliary storage medium is a magnetic drum, capable of storing approximately 10,000 words. As is typical of auxiliary storage, orders stored on the drum cannot be executed and the operation of transferring the contents of the drum to the Williams memory is time-consuming (1000 words per second).

ILLIAC's role in the Cornfield System is to control. Input to ILLIAC consists of the contents of the tracking computer memory and manually inserted instructions; output consists of instructions for controlled air traffic. As shown in Fig. 2, a direct link connects ILLIAC output to TASC. This link simulates a digital data link to controlled air traffic (simulated targets

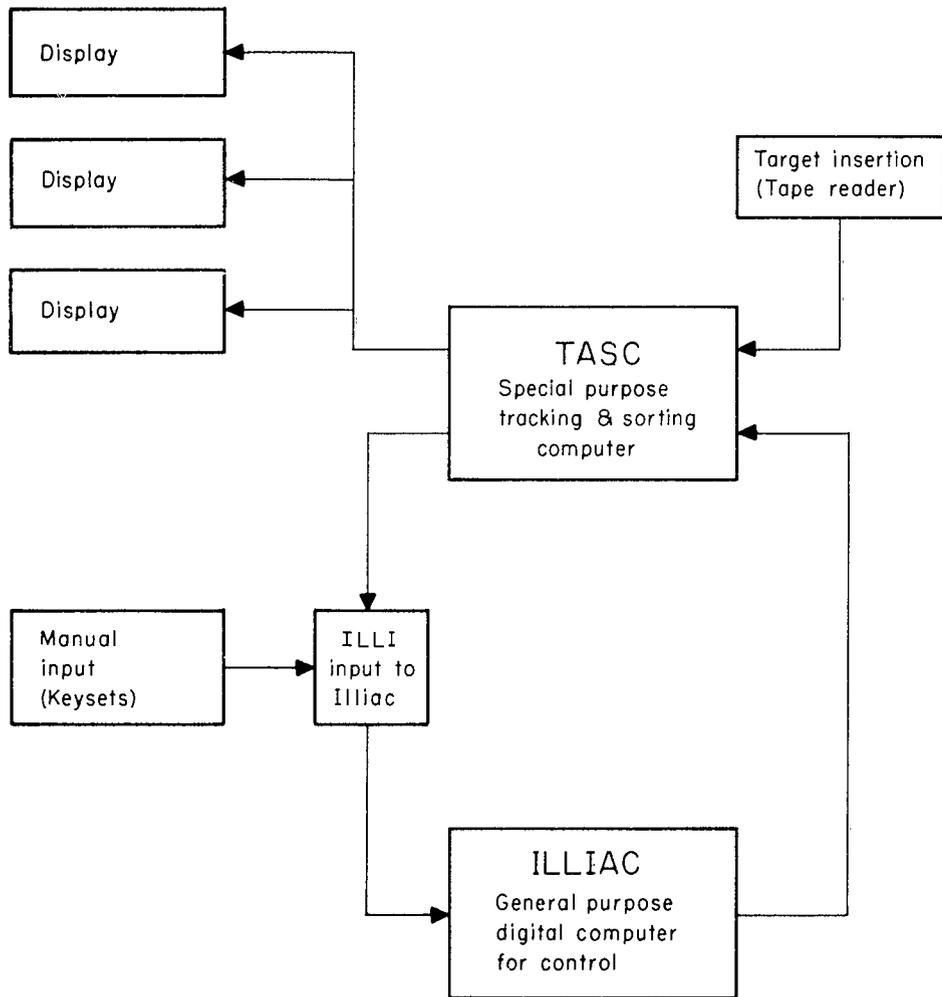


Fig. 2. Cornfield System

reside in TASC). Being a general purpose machine, ILLIAC can be programmed to control as any experimenter chooses.

2. Tracking and Sorting Computer

TASC is described in detail in CSL Reports R-35, R-102, and R-114. Briefly, the machine is a wired-logic, digital device employing a drum for storage of tracks. It can accept digital (not video) radar data at a rate of 100 reports per second and can automatically track 102^4 targets in real time. It also has auxiliary input channels which convey information from manually operated keysets, a paper tape reader, and ILLIAC. The machine has two output channels; one transmits target information in parallel (104 parallel bits), and the other transmits target information serially in bits (80 bits). The parallel channel feeds ILLIAC and the serial channel drives the displays.

The role of TASC in the work described herein is to 1) simulate air traffic, 2) provide storage for auxiliary information descriptive of targets and 3) to drive the displays. Since TASC is used to simulate targets, the radar input channel has been disconnected. Target simulation by TASC is equivalent to perfect tracking described in more detail later in the report. Airports, runways, outer markers, etc. are also stored on the TASC drum as stationary targets.

3. Display

The serial output channel of TASC drives three displays. Each display consists of a 19 inch charactron (C19K) with P-19 long persistence phosphor. All displays incorporate scale selection and off-centering so that each display can be switched to display en route area, approach area or terminal area as desired. Targets appear as spots with velocity noses, and auxiliary

information appears as alpha-numeric characters beside each target. An operator has the ability to filter the information available for display by manipulating toggles. In addition, airports, runways, outer markers, etc. are displayed as symbols.

In the experimental work described herein, the displays were not used in simulation of the display channel described in Part I of this report (see Fig. 1), but were used as a means for observing traffic behavior in experiments. Referring to Fig. 1, the reader will see that the displays of the projected system are driven by a set of display logic intended to provide a necessary link between operators and the automatic system. The logical properties of this link were not a part of the study described in this report.

4. ILLIAC Input

ILLI is a logical device which serves to expand ILLIAC's input ability. ILLI has fifteen input channels and one output channel.

5. Manual Input to ILLIAC

Manual input to ILLIAC is by keyset. Several different keysets are used, ranging from keyboards for insertion of program changes and parameters, to small keysets for rapid insertion of instructions. The latter keysets were used extensively for experiments in which controllers were used to control air traffic.

6. Target Insertion

Targets are inserted through a paper-tape reader (Fig. 2), which can be switched to run from a clock, providing automatic insertion of targets.

The tape containing the targets is referred to herein as the "script." The script determines traffic loads and configuration, and arrival times of simulated air traffic.

7. Limitations

The Cornfield System has two disadvantages which have had direct bearing on the results of this work: 1) ILLIAC's active memory is too small, and 2) the TASC-to-ILLIAC data link is too slow. In spite of our efforts to minimize the effects of these two factors, they have caused some degradation in performance and have established a system capacity of 25 aircraft. Both problems are non-existent in the projected system described in Part I.

ILLIAC Memory. The final version of the automatic control program has a length of 1500 words. Since the Williams memory consists of 1000 words, it was necessary to resort to drum storage, with two 500-word drum transfers every 3 seconds. Consequently, $1/3$ of the time available for control computation has been lost to drum transfers.

TASC-ILLIAC Data Link. TASC transmits one target to ILLIAC every 2.5, 5.0, 7.5 or 10 milliseconds, depending on target distribution on the TASC drum. If ILLIAC is busy at the time TASC transmits, a time interval of some integral multiple of 2.5 milliseconds will elapse before TASC will again transmit a target. In operation, ILLIAC is frequently busy, partly because it is relatively slow in computation, and partly because it is slow in reading (approximately 2 milliseconds per target). These factors determine a target access time. In the projected system described in Part I, the same access time is a core-memory access time, typically 2.5 microseconds.

B. Assumptions and Restrictions

All assumptions and restrictions fall into one of three categories: 1) in air traffic, 2) in environment, and 3) in the control system. Two automatic control programs have been assembled, one (PICON) for use in studying approach and terminal sequencing control, and the other (PICON-ER) for use in studying en route control. The two control programs differ only in assumptions in environments. In addition, a third program (PILOT) has been assembled to simulate the intelligence in the cockpit of a typical controlled aircraft. PILOT makes up a significant part of the assumptions in air traffic as described below.

1. Assumptions in Air Traffic

Acceleration. All aircraft accelerate at the same rate, namely 18.7 mi/hr/3 sec linear, and 3° /sec in a turn (standard needle width turn).

Speed. There are four types of aircraft, each defined by a speed range as listed below:

<u>Type</u>	<u>Min. Speed</u>	<u>Max. Speed</u>
0	112 mph	131 mph
1	131 mph	206 mph
2	168 mph	355 mph
3	206 mph	542 mph

Emergency. In-flight emergencies cannot occur. Emergency procedure is trivial to program, and so was not included in this work, in the interest of conserving memory space.

Well-behaved Aircraft. By definition, a flight is well-behaved if headings are accurately held, and linear acceleration is zero, except as determined by PILOT logic. Poorly-behaved traffic is discussed in the section describing control logic.

PILOT. This program was designed to simulate the pilot-aircraft combination to the extent necessary for experiment. As such, it adds two important properties to simulated air traffic: 1) the ability of each simulated flight to execute instructions, and 2) the ability of each simulated flight to make arbitrary decisions. In the work described herein, only sensible arbitrary decisions were allowed, such as adjusting speed, or turning to maintain a track along the ILS. It was desired to include a set of non-sensible arbitrary decisions (such as unstable headings and arbitrary speed changes) but this was beyond the capability of the Cornfield System.

PILOT is a complete program in itself and can therefore be used in experiments in manual control. If used for this purpose, control instructions come from controllers (via keyset) and the simulation is, in effect, a controller, equipped with voice link to all controlled aircraft. Many such experiments have been run, their primary purpose being to reveal and test techniques of control for use in the automatic program.

Regardless of whether PILOT is used in manual control or as part of the automatic control program, it has the capability of executing the following instructions:

- 1) Vector θ : Cruise on a heading of θ , turning if necessary.
- 2) Hold θ : Enter a standard 4 minute holding pattern (left) such that the final one minute leg of the pattern is on a heading of θ .
- 3) Vector Final θ : Turn (if necessary) to a heading of θ and expect to contact the localizer; upon localizer contact, proceed without further instruction to execute an instrument landing (subject to missed-approach decision).
- 4) Take-off

The arbitrary decisions (listed below) made by PILOT are intended to resemble those a real pilot would make. All speed changes are made at the stated rate.

- 1) Reduce speed to slow cruise in a holding pattern.
- 2) Observe an approach area speed limit.
- 3) Reduce speed to slow cruise while in terminal phases of flight.
- 4) Make the missed-approach decision, upon passing the outer marker; PILOT decides to miss if a flight is poorly lined up or is traveling too fast as it crosses the outer marker.
- 5) In all other cases, increase speed to normal cruise.

2. Assumptions in Environment (PICON)

Winds Aloft. Winds aloft are zero.

Altitude. One altitude exists.

Control Area. En route control area is a square 128 miles on each side; there is only one control area.

Airport. An airport consists of one runway with ILS (including outer marker), terminal sequencing area with 4 entry points, and an approach area. Entry points are geographical points approximately 5 miles from the outer marker (Fig. 11). Approach area is circular, 30 miles in radius, with center of the circular area at the outer marker.

Number of Airports. One airport exists within the simulated en route control area.

Arrival. An arrival occurs when a flight reaches the far end of the runway (upwind end).

Approach Control. One approach-control computer exists. In PICON, the assumed machine executes approach-area control and terminal-sequencing control.

3. Assumptions in Environment (PICON-ER)

Winds Aloft. Winds aloft are zero.

Altitude. One altitude exists.

Control Area. En route control area is a square, 512 miles on each side; there is only one control area.

Airport. An airport consists of an approach area 30 miles in radius and a terminal sequencing area 5 miles in radius. The two areas are concentric.

Number of Airports. A maximum of sixteen airports exists in the en route control area.

Arrival. An arrival occurs when a flight reaches terminal sequencing area.

4. Assumptions in the Control System

Tracking. Automatic tracking is perfect. Even in the highest quality automatic tracking system, there will be an occasional "lost target." This event can occur if a target's position is not reported for several minutes or if a flight arbitrarily makes a sharp turn. It is easy to detect the event and also easy for a controller to correct the situation. In this work, it is assumed that the event can not happen.

Tracking Computer Error. Errors in position information are small with respect to minimum controlled spacings. (Minimum controlled spacing is 8 miles in en route area and 1 1/2 miles in terminal area).

Identification. All controlled traffic enters the system fully identified. Since identification is a manual process (as described in Part I) and the study was not intended to simulate manual processes, identification is assumed to have occurred prior to target entry into the control area.

Number of Flights. Maximum number of flights at any time is 64.

Number of Systems. Only one automatic air traffic control system exists. The automatic communication (center to center) was not simulated since the Cornfield System has insufficient capacity.

Communication. The automatic control computer can transmit control information to any controlled flight via digital data link. Information content is 1) identity of the flight addressed, 2) heading in degrees, and 3) control status (hold or cruise).

III. AUTOMATIC CONTROL

In this work, control is divided into two main components: 1) basic logical functions and 2) control logic. The basic logical functions can be regarded as the tools with which the control logic is executed. An effective and powerful set of tools facilitates a flexible and effective control logic; conversely, if a control logic is specified, tools must be devised such that the logic can be executed in a digital machine. The tools and the logic are equally important and dependent on each other.

An important factor to be considered in the design of an automatic system is the use of restricted air space. By definition (in this report) air space is "restricted" if traffic is confined to specified air space. Restriction of air space results in a mixture of advantages and disadvantages depending on how it is used. In the existing manual system, for example, extensive use of airways has the undesirable effect of generating conflicts at VOR stations where airways must converge. On the other hand, the establishment of the high-altitude positive-control area has the advantage of defining a restricted air space in which only controlled traffic can operate. In this work, en route air space is totally unrestricted. Therefore controlled traffic is not confined to specified tracks or airways, but is instead confined to specified headings. The purpose and the advantage of this is to make more efficient use of air space.

In the terminal area, the presence of a fixed runway specifies a final approach path that is certainly restricted air space. Because of the high accuracy necessary in following the approach path, it is not practical to maintain traffic "restricted to heading" throughout final approach. As in the existing system, the transition from heading-following to fixed-track-following is to be made at the gate, a point about 3 miles upwind from the outer marker.

PICON has been designed so that landing traffic will cross the gate aligned with the runway and within 1/2 mile of the localizer center-line.

In approach area, the degree of restriction of air space lies between the two extremes mentioned above. In PICON, approach area traffic is not restricted to fixed tracks, but is restricted to one of 4 pie shaped sectors. As far as pilots are concerned, however, the restriction is to a heading, and this is true of all phases of controlled flight, except for final approach.

A. Basic Logical Functions

The basic logical functions of the special purpose machine depicted in Fig. 1 are analogous to the order code of a general purpose digital computer. Where the general purpose machine has ample need for such functions as "add," "subtract," "multiply," etc. the special purpose machine requires a set of functions chosen to solve the special problems of air traffic control. A few of the more obvious functions required in the special purpose machine are: calculate the distance between two given aircraft, calculate a heading to destination, calculate distance to destination, etc. In all of these functions, arguments would either reside in the memory or in accumulators.

The following is a description and in some cases, an evaluation of the more important basic logical functions used in PICON. Most of these are subroutines. Those that are not, may more accurately be called "basic logical principles" that were used in PICON.

1. Sort

"Sort against" is a term used in this report to indicate a special type of comparison and selection. In all cases, a single item is compared with many other items, and the comparison is special, because several different checks or comparisons can be made in one sort. It is proper, therefore, to sort a reference aircraft against all others, making a distance comparison in the sort so that at the completion of the sort, the aircraft nearest the reference aircraft shall have been selected. The reference item need not be an aircraft, however; it could be an airport, for example. A sort with respect to an airport would yield the aircraft nearest the airport.

Sorting is a technique that is a permanent part of any automatic control system, since it is the only way digital equipment can find answers to such questions as "Which flights are near?" "Which flight will arrive first?" etc. The process is obviously time-consuming, since it implies that for 1000 aircraft, it will be necessary to continually sort one against 999 others. Several methods are available for reducing sorting time, however, some of which are described in detail later (ΔJ_1 Logic).

2. Reference Aircraft Concept

There are at least two general ways of attacking the problems of controlling many aircraft, one of which consists of solving n equations in n unknowns where $n \leq 1000$. This method is relatively unwieldy, particularly in terms of the Cornfield System limitations. The other method, used in PICON, is described below.

The reference aircraft concept is one in which all control problems are solved for a selected reference aircraft, under the assumption that all other flights cannot be changed. Two problems are involved: 1) how to

select the reference aircraft and 2) how to design a system of logic such that conflicts can always be resolved by manipulating only a reference aircraft. In PICON, selection of reference aircraft is determined by nearness to destination and proximity of aircraft. The second problem listed above has not been completely solved; it is described later.

The reference aircraft concept is very likely a permanent part of an automatic control system. When used in conjunction with a proximity condition, it has the advantage of reducing the worst-case conflicts to known configurations of not more than 5 or 6 aircraft.

3. Pairing Rule

The pairing rule is based on the following: if a reference aircraft is in conflict with more than one other aircraft, it is adequate to avoid only the nearest conflict; this is true because all other conflicts must lie along the original path of flight of the reference aircraft. By the pairing rule, only the nearest conflict is selected for avoidance and consequently, the most complicated air situation is reduced to a pair of flights consisting of the reference aircraft and its nearest conflict. If there should be a third aircraft nearby, it will not be taken into account, until, or unless, it becomes the nearest conflict. The same is true of a fourth, a fifth, etc.

Out of necessity, the pairing rule was used both in conflict detection and in conflict resolution in PICON. In conflict detection, it has the advantage of simplifying the computations. In resolution, it has the disadvantage of ignoring secondary conflicts. (The primary conflict is the nearest conflict and a secondary conflict is one generated in turning the reference aircraft to avoid a primary conflict). Even though a secondary conflict will eventually become a primary conflict and subsequently be avoided, the rule is

probably inadequate in conflict resolution. The ramifications of the pairing rule in conflict resolution have not yet been fully explored.

4. Class Switching

In PICON, ten classes of flight were defined, along with ten separate sub-systems of logic. The class of each flight is chosen and set by control, depending on the phase of flight or the predicament of the flight. The class, in turn, determines which of ten sub-systems of logic shall be used in processing each flight. The following is a table of the classes employed in PICON.

<u>Class</u>	<u>Phase of Flight</u>
V (vector)	en route, normal cruise
VC (vector conflict)	en route, in conflict
H (hold)	en route, holding
AC (approach control)	approach area, no conflict
HA (hold approach)	approach area, holding
HO (hold terminal)	approach area, holding for entry to terminal area
VF (vector final)	terminal sequencing vector to ILS
P (maneuvering)	turning into localizer
L (landing)	final approach between outer marker and turn off the runway

Class switching is a powerful function and will certainly be a part of any automatic control system. It has the advantage of reducing a huge control problem to several smaller problems each of which can be solved independently. In PICON, the class switch was programmed, but in a special purpose machine, much more efficient and time-saving switching can be obtained with an electronic device. Though PICON uses ten classes, it appears that 15 or 20 would be more desirable in a real system.

5. Variables

In PICON, variables were chosen so that the same variables could be used in conflict detection, conflict resolution, and minimization of computation time. Three variables are defined to measure 1) proximity, 2) future proximity, and 3) time to nearest passage. All three quantities are derived from the general expression

$$t_{ci} = \frac{d(t)}{s_1 + s_i}$$

in which $d(t)$ is the distance between the reference aircraft and the i th aircraft, s_1 is the speed of the reference aircraft, and s_i is the speed of the i th aircraft. A detailed derivation of each variable is given in Appendix A.

Proximity. Proximity is defined as the time interval

$$t_{ci} = \frac{d(0)}{s_1 + s_i}$$

where $d(0)$ denotes the distance at the time of calculation. Since s_1 and s_i are not vector quantities, t_{ci} is the time interval necessary for the reference aircraft to have a head-on collision with the i th aircraft. Re-stating in a more useful way: the reference aircraft cannot possibly have a collision with the i th aircraft in a time less than the interval t_{ci} . If the collision sort, against all i , selects the smallest t_{ci} ($t_c = \text{smallest } t_{ci}$), then it is true that the reference aircraft cannot possibly collide within t_c and therefore, the safety of the reference aircraft is assured over the next time interval of t_c . This statement is true regardless of air traffic density or orientation, for well-behaved traffic. (In the real case, air traffic is not well behaved, of course. This factor only has the effect of reducing the time interval of safety, it does not change the logic described above. This point is expanded in detail in the following pages.)

Time to Nearest Passage. The time interval, t_{mi} , is defined as that which will elapse before the reference aircraft and the i th aircraft reach minimum separation if present headings and speeds are maintained. The smallest t_{mi} is defined as t_m ; this is the time interval that will elapse before the reference aircraft has a "nearest passage."

Future Proximity. Future proximity is defined as

$$t_{cmi} = \frac{d(t_{mi})}{s_l + s_i}$$

The interval, t_{cmi} , is the smallest value of t_{ci} that will exist if the reference aircraft and the i th aircraft maintain present headings and speeds. The smallest t_{cmi} is defined as t_{cm} ; this quantity is the smallest value of future proximity that the reference aircraft will experience, if its present heading and speed is maintained.

The three variables described above are all intervals of time. If t_0 is the present real-clock time, then the reference aircraft cannot possibly collide with any aircraft before $t_0 + t_c$, by the real clock. At $t_0 + t_m$, by the real clock, the reference aircraft will be at its closest approach to another aircraft and at that time, a collision will be impossible before an additional time interval of t_{cm} has passed. The effects of non-well behaved aircraft are discussed under the Collision Sorter.

6. ΔJ_1 Logic

Regardless of the methods used, the task of controlling up to 1000 aircraft simultaneously in a digital computer will be time consuming. This fact is accentuated when one observes that each flight will have to be periodically sorted against all the others. In spite of the high computation

speeds available today, it is desirable (perhaps even necessary) to seek methods of making more efficient use of digital machines, in anticipation of excessive computation times. In air traffic control, calculation cannot be avoided but considerable flexibility exists in the frequency of calculation. If one asks "How often is it necessary to sort one aircraft against all of the others?" the best answer clearly depends on the proximity of other aircraft or nearness to destination or both. With this idea in mind, the following logic was incorporated in PICON.

Each aircraft has associated with it a real-clock time, J_i , which is to be calculated and stored anew each time that the i th aircraft is a reference aircraft. This time is formed by the following relationship:

$$J_i = t_0 + \Delta J_i,$$

where t_0 is the present clock time, and ΔJ_i is the time interval over which the i th aircraft does not need control processing. As each aircraft is fed into the control computer for processing, its J_i is tested as follows:

if $J_i > t$, do not admit for processing;
 if $J_i \leq t$, admit for processing;

where t is the clock time. Since real-clock time is continually increasing, the i th aircraft will eventually have $J_i \leq t$ and be admitted for processing. At the end of processing, the control computer must form a new J_i from the information gathered and calculated in processing. The new J_i is stored, and in the future compared again with t , thus completing the cycle.

Formation of J_i must be such that the i th aircraft is processed often as it approaches its destination, and often as it nears other flights, whether

in conflict or not. In all other cases, the rate of processing can be some minimum provided that any possible conflict is anticipated. These conditions are met if

$$\Delta J_1 = \min[k_1 t_c, k_2 t_d, T_k]$$

where t_d is the time interval for the reference aircraft to reach its destination, T_k is the absolute maximum time interval over which it is safe to avoid processing any aircraft, and the parameters k_1 and k_2 are chosen as described below.

The selection of the smallest of three time intervals for ΔJ_1 is a selection of the most important event influencing the reference aircraft. If $k_1 t_c$ is smallest, then the influence is nearby aircraft; if $k_2 t_d$ is smallest, then nearness to destination is most important (if $t_d < t_c$ then the flight cannot possibly have a collision before reaching its destination). If T_k is smallest, then the reference aircraft is not near its destination or another aircraft and the choice of T_k for the non-processing time interval can be based on other factors described below.

$k_1 t_c$ Smallest. As mentioned above, if $k_1 t_c$ is the smallest of the three quantities, then the most important influence on the reference aircraft is another aircraft. If 1) control has the guaranteed ability to resolve any conflict and if 2) ΔJ_1 is chosen less than t_c , then the extreme low probability "turn and proceed to head-on collision" is fully anticipated. Furthermore, the only price paid for full protection against this event is a higher processing rate for pairs of aircraft passing near each other. The first condition above can be met by carefully designing the control logic and the second condition is met by selecting k_1 so that $0 < k_1 < 1$.

In PICON, k_1 was set to $1/2$ and held fixed (holding k_1 constant is adequate if traffic is well-behaved and wind aloft is zero). The effect of setting k_1 to $1/2$ is to guarantee that if two aircraft should turn to head-on collision headings immediately after processing, one or both will be processed after half the time interval to collision has passed. If they do not turn to head-on collision courses, they will be processed in the same time interval as if they had turned. This has the desirable effect of processing each flight more often while in the vicinity of other flights (whether in conflict or not), and not so often as flights become more isolated. This effect is, of course, independent of the value of k_1 .

In a real system, k_1 would be a variable. It would be some function of 1) safety factor, 2) errors in the system, 3) type and speed of aircraft, and 4) wind aloft. Safety factor and errors in the system would set the upper limit of k_1 . Use of a safety factor in k_1 does not imply that safety is a variable in a normal operation of aircraft. Safety is a variable, however, in the extreme case of two flights arbitrarily turning to head-on collision courses. (If k_1 were set to unity and the low-probability event did occur, the computer would discover the event at the moment of collision). Errors in the system may well vary from area to area, and even from radar to radar; this variation should be reflected in the upper limiting value of k_1 .

Use of type and speed in determining k_1 is best illustrated by example. If two controlled aircraft known to be jet airliners were passing near each other, the probability of the dangerous turn is quite low, and the upper limiting value of k_1 provides adequate safety. If, however, one of the jets were cruising at 200 mph, k_1 should be reduced, thereby increasing the processing rate in anticipation of the low-probability event aggravated by

a subsequent speed increase. The price paid for full anticipation of the low-probability event is only a temporary increase in the processing rate.

The effect of non-zero winds aloft should cause k_1 to increase or decrease depending on whether the dangerous turn, as affected by the wind, will increase or decrease the time to collision.

It is conceivable that even more efficient use could be made of control computer computation time by deducing types of aircraft in the uncontrolled population. If, for example, an uncontrolled flight has been holding a heading and maintaining 200 mph for a relatively long period of time, the flight would very likely be a propeller driven craft with a top speed of 220 mph. If such logic were incorporated, it would have the effect of buying computation time and thereby increasing system capacity. The safety of controlled traffic would be jeopardized only if the type detecting logic were "tricked" by a high-performance aircraft, exhibiting low-performance characteristics for a long period of time.

As mentioned earlier, k_1 was set to $1/2$ in PICON. Although it is not difficult to simulate wind and poorly behaved traffic, it was not done in PICON because of the system limitations described earlier.

$k_2^t t_d$ Smallest. If $k_2^t t_d$ is the smallest of the three quantities, then the most important factor influencing the reference aircraft is nearness to destination. In PICON, k_2 was set to $1/2$. Very likely nothing can be gained by considering k_2 as a variable. If k_2 is simply $1/2$, the effect is to process a controlled flight each time the flight has traversed half the distance remaining to destination. This has the desirable effect of processing each flight more often at the most critical time, i.e., as the flight approaches terminal phases of flight.

T_k Smallest. If T_k is the smallest quantity of the three, then the controlled flight is not near its destination or near any traffic. The factors entering into choice of T_k are mostly system considerations although wind aloft may have some effect.

If T_k is chosen so large that a flight can traverse an entire en route control area without control processing, there arises a problem of forwarding its control instructions. If T_k is chosen so that the flight must be processed several times in traversing an en route control area, then no special steps need be taken to forward control instructions, since forwarding is part of processing.

Winds aloft enter into the choice of T_k only if air traffic is used to determine the wind. If T_k is the smallest of the three quantities, then $T_k < k_1 t_c$, and the nearest aircraft would be too distant to be of any value in determining the wind for the reference aircraft. To avoid devious routing due to strong unknown winds aloft, it may be necessary to let T_k be some function of the time since winds aloft were last computed.

7. Collision Sorter

The technique of collision sorting in PICON is excessively time-consuming because of Cornfield System limitations. In a system such as that described in Part I of this report, collision sorting equivalent to that described here could be accomplished in times that are 3 orders of magnitude less than required for PICON. The reasons for this are given in detail in Summary and Conclusions. The flow chart for the collision sorter of PICON is shown in Fig. 3.

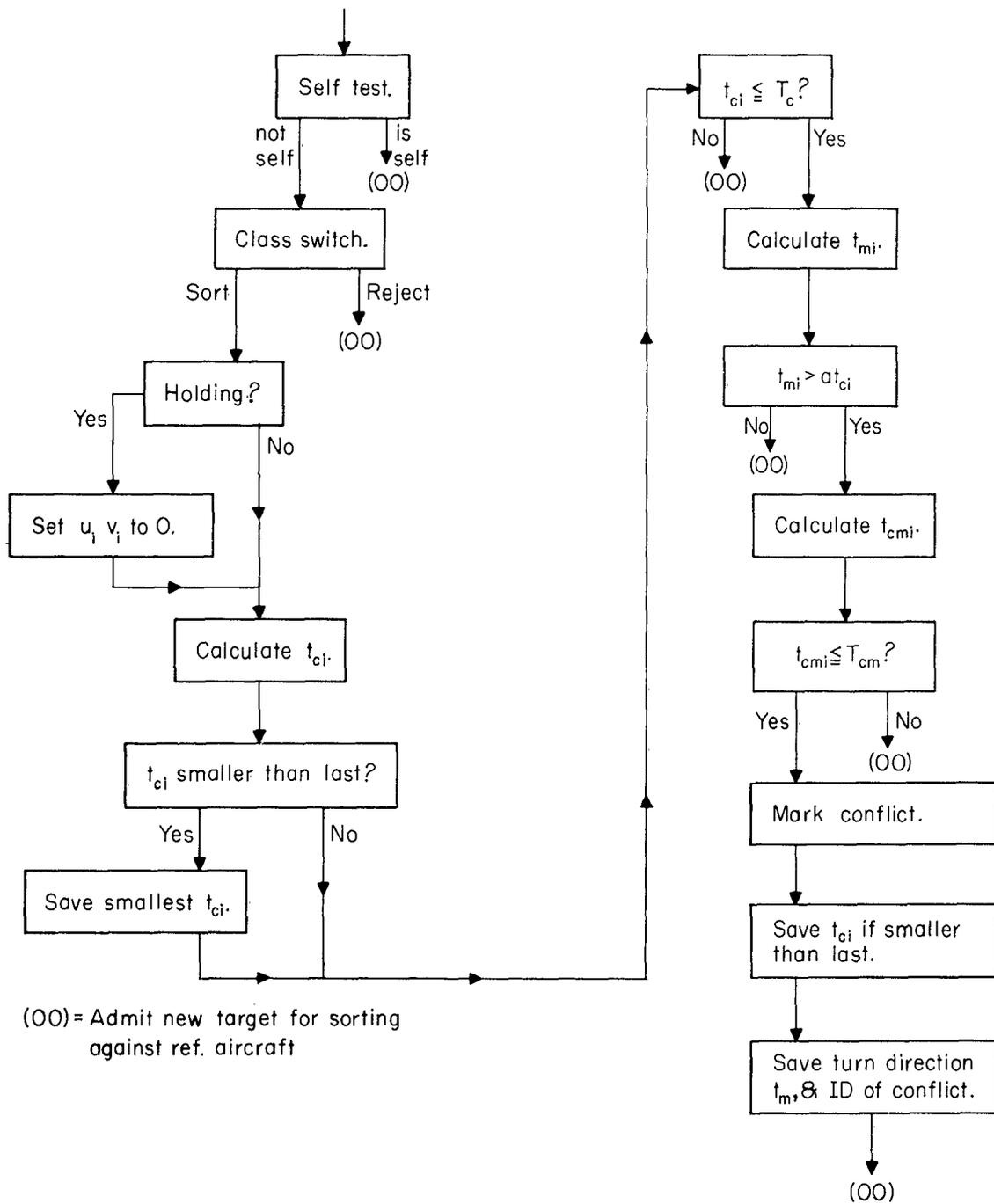


Fig. 3. Collision Sorter Flow Chart

The function of the collision sorter can be stated as follows: given a reference aircraft and the conditions for the sort, compare all aircraft with the reference aircraft in several different ways, rejecting those that do not meet certain conditions (described below). At the conclusion of the sort, sorter storage must contain all information of immediate influence on the reference aircraft.

In PICON, the collision sorter has been written as a subroutine that can be entered and executed in the same way as any other subroutine. Entry to the sorter causes ILLIAC to set aside the next full TASC frame for collision sorting. (A TASC frame consists of every target in the system). Even though much time has passed and an entire frame has been used in the sort, exit is the same as for a standard subroutine. This point is mentioned to facilitate reading the flow diagrams showing control processing. The subroutine has three conditions for exit: 1) the collision sorter is busy, i.e. a reference target has been selected and is awaiting the sort frame, 2) the subroutine was successfully entered and the target presently being processed has been selected as the reference aircraft, and 3) the collision sort is completed and all pertinent information is stored in sorter storage.

The subroutine can also be instructed to sort with any one of three different headings for the reference aircraft as follows: 1) use the present heading (θ_p) of the reference aircraft in the sort; this is normally used when it is intended that the reference flight shall continue on its present heading (as in conflict class VC), 2) use the reference aircraft's heading to destination (θ_s) in the sort, ignoring the present heading; this is normally used in sorting holding flights (class H); and 3) use given heading (θ_g) which may be different from the other two cases; this type of sort was intended for use in special situations.

Holding Flights. Sorting against a holding flight presents a special problem because the heading of a holding flight sweeps the entire compass once each circuit of the holding pattern. This has the effect of creating pseudo conflicts for the reference aircraft while, in fact, that which the reference aircraft must avoid is the holding pattern itself. In PICON, the problem was solved by approximating the holding pattern as follows: Each time the i th aircraft is a holding flight, the sorter will set u_i and v_i to zero but will retain s_i . (The quantities u_i , v_i , and s_i are x and y components and magnitude of velocity, respectively). Having made these changes, the sorter will proceed with all calculations using the modified values. In effect, this establishes the hold-pattern position as that occupied by the holding flight at the instant of calculation; it also defines a circular avoidance area centered on holding position, and of radius proportional to $s_1 + s_i$. Clearly the ideal avoidance area is not circular (it is shaped like a race track); yet the ideal avoidance area must be avoided. Therefore, the circle actually avoided has been enlarged to maintain safety in exchange for efficiency.

8. Conflict Detection

Rejection of non-threatening aircraft is an inherent part of the purpose of the collision sorter as stated previously. If it is possible to reject a flight without going through all of the calculations necessary to determine threat, it should be done in the interest of saving computation time. The comparisons and decisions have been arranged so as to cause early rejection in the majority of cases.

Class Filter. The class filter is a first-order rejection of traffic that cannot possibly be in conflict with the reference aircraft. If, for example, the reference aircraft is in one of the en route classes, it could not possibly be in conflict with a flight in landing class. The following table lists the class filtering used in PICON.

<u>Class of Ref. Aircraft</u>	<u>Sort Against</u>	<u>Reject</u>
V (vector, en route)	V VC H AC HA	HO VF P L
VC (conflict, en route)	same as V	same as V
H (hold, en route)	same as V	same as V
AC (approach area)	AC HA HO	V VC H VF P L
HA (hold approach)	same as AC	same as AC
HO (terminal sequencing)	not sorted	not sorted
VF " "	"	"
P " "	"	"
L " "	"	"

Class filtering is an important concept and would very likely be a part of the real system described in Part I. It was, of course, necessary to program class filtering (in the program it is an extension of the class switch). Electronic devices could do the job of class filtering in much less time than consumed in PICON.

Proximity Condition. The next level of rejection is still a crude one intended to reject all traffic too distant to be of any concern. The collision sorter calculates t_{c_i} for use in a decision to reject the i th aircraft or continue processing as follows:

if $t_{ci} > T_c$, reject the i th aircraft;
 if $t_{ci} \leq T_c$, proceed to the next calculation.

In the above expressions, T_c is the proximity parameter and is set to 15 or 20 minutes. The main factors influencing the choice of T_c are: 1) the length of time the control computer needs to resolve the most complicated traffic situation and 2) the value of k_1 . This is best illustrated by example: assume $T_c = 20$ minutes, $k_1 = 1/2$, and an air situation such that for a given reference aircraft, t_c is slightly greater than 20 minutes. By the proximity decision given above, the reference aircraft will be declared not in conflict, and control will set ΔJ_1 to $k_1 t_c$ (10 minutes). Ten minutes will elapse before the situation is again examined by control. If the reference flight was indeed on a collision course with another aircraft, the conflict will be discovered on the second examination with 10 minutes of maneuvering time remaining.

Elapsed Time to Nearest Passage. If a target is not rejected in the proximity decision, it will be used to calculate t_{mi} for the reference aircraft. The quantity, t_{mi} , is then used in the following decision:

if $t_{mi} < 0$, reject the i th aircraft;
 if $t_{mi} \geq 0$, proceed to the next calculation.

These statements are derived as follows: if t_{mi} is negative, nearest passage occurred in the past, therefore, at present, the reference aircraft must be on a heading that is divergent with the i th aircraft. If t_{mi} is zero or positive, the time of nearest passage is at present or in the future, and conflict is possible.

Future Proximity. If the nearest passage is to occur in the future, the i th aircraft is retained for use in calculating future proximity with the reference aircraft. The future proximity variable, t_{cmi} , is calculated and used as follows:

if $t_{cmi} > T_{cm}$, reject the i th aircraft;
 if $t_{cmi} \leq T_{cm}$, the reference aircraft is in conflict with the i th aircraft.

In the above expressions, T_{cm} is the future proximity parameter and is normally set to 2 minutes. If, at the conclusion of a collision sort, $t_{cm} \leq 2$ minutes, then the reference aircraft is in conflict (by definition), and must be steered accordingly. Converting T_{cm} (of 2 minutes) to distance, it is found that no conflicts will exist if future proximity is greater than 40 miles for two jets, greater than 8 miles for two light aircraft, and greater than 24 miles for a jet and a light aircraft.

It is desired to choose T_{cm} as small as possible, of course, to increase utilization of air space. The minimum safe value of T_{cm} is determined by such factors as radius of turn of aircraft, tracking computer delay in tracking turns, and the ability of control to resolve special conflicts. Special conflicts (not discussed in this report) are those in which conflicting flights are so close to each other that conflict resolution becomes contingent upon radii of turns. All of the above factors stem from the assumption that control must be able to resolve the low probability event in which both flights arbitrarily turn to head-on collision courses.

In summary, the following three conditions must be met to establish a conflict for the reference aircraft:

$$1) t_c \leq T_c$$

$$\text{AND } 2) t_m \geq 0$$

$$\text{AND } 3) t_{cm} \leq T_{cm}.$$

Conversely, the reference aircraft is not in conflict if one of the following conditions prevails:

$$1) t_c > T_c$$

$$\text{OR } 2) t_m < 0$$

$$\text{OR } 3) t_{cm} > T_{cm}.$$

9. Conflict Resolution

The main problem in conflict resolution is not so much avoidance of the primary conflict, as it is the generating of secondary conflicts as a result of turning to avoid a primary conflict. The pairing rule particularly aggravates this problem since it guarantees that a secondary conflict will not even be discovered until it becomes the primary conflict. This problem can only be properly handled with another sorter which might be called the resolution sorter. The structure of a resolution sorter is so different from the collision sorter that the collision sorter could not be modified to serve the purpose. Instead, the collision sorter itself was used, with considerable compromise. In effect, dual use of the collision sorter constitutes a trade, in which memory space is gained and computation time is lost.

Iterative Process. A conflict class (VC for vector conflict) was defined along with a system of conflict logic. As its name implies, any flight in VC class is in conflict and the conflict logic shall be such as to resolve the conflict. The method is as follows: If a non-turning flight appears in VC class, logic of VC shall instruct the target to turn to a heading 180° different from its present heading. Every three seconds thereafter, the flight shall be collision sorted using θ_p (present heading) until the conflict has disappeared. Since the flight will turn at a standard rate, the effect is to check for conflict after each 9° increment of turn. The method has two advantages: 1) it makes efficient use of limited memory space since it uses existing sub-routines (collision sorter) and 2) it takes into account turn radius, which is a relatively complicated calculation. There also are two disadvantages in the method: 1) it requires a time-consuming collision sort every three seconds for each aircraft in conflict and 2) the method sheds no light on the problems of design of a real machine because it cannot be used in a real system. This is further explained in the next paragraph.

The iterative method is an acceptable method of simulation because it yields the same answers in heading correction as a direct calculation would. However, the success of the method is determined exclusively by knowledge of position and heading of turning traffic. In the Cornfield System, perfect tracking is assumed, and therefore positions and headings are known to a satisfactory accuracy throughout a turn. In a real system, it is probably impossible to have such knowledge on turning aircraft because of the unpredictable lags, both in pilot response, and in computer ability to track a turn. Aside from this, the time consumed in calculation would be prohibitive. It is conceivable and practical to devise a system of arithmetic that will simply calculate a no-conflict heading for transmission to the affected aircraft.

Quantization Problem. This is a problem in conflict resolution and is very likely a permanent part of an automatic control system using digital data in its calculations. The problem stems from use of digital data and not from any particular system of logic.

The tracking computer is a digital computer with a limited number of position and velocity digits. Because of this, the computer's representation of aircraft position will rarely coincide exactly with the true position. Instead, the computer's position report will always be one of four corners of a tiny bin surrounding the aircraft. The dimensions of the bin are the smallest unit of distance that the computer can carry, and the quantization error is always some fraction of the dimensions of the bin. In the Cornfield System the dimensions of the bin are 1/2 mile on each side.

In PICON, a conflict exists if

$$1) \quad t_c \leq T_c$$

$$\text{AND} \quad 2) \quad t_m \geq 0$$

$$\text{AND} \quad 3) \quad t_{cm} \leq T_{cm}$$

Conversely, no conflict exists if

$$1) \quad t_c > T_c$$

$$\text{OR} \quad 2) \quad t_m < 0$$

$$\text{OR} \quad 3) \quad t_{cm} > T_{cm}$$

Given a conflict, the problem is to find a heading (θ_{nc}) such that the conflict disappears or, in other words, find a θ_{nc} such that any one of the three "no conflict" conditions is met. Taking each of the three cases separately, it is seen that

- 1) θ_{nc} such that $t_c > T_c$ is impossible,
- 2) θ_{nc} such that $t_m < 0$ is arbitrary,
- 3) θ_{nc} such that $t_{cm} > T_{cm}$ is a heading that will allow

passage at minimum-safe separation.

Having resolved a conflict by finding θ_{nc} such that $t_{cm} > T_{cm}$, the two flights shall proceed to the point of nearest passage without entering into conflict again (well-behaved aircraft assumed). At the point of closest passage, however, a pseudo-conflict can occur due to quantization error in $d(0)$. If the error in $d(0)$ is such that $t_{ci} \leq T_{cm}$, then there is no θ_{nc} such that $t_{cm} > T_{cm}$ since $t_{cm} = t_{ci} \leq T_{cm}$.

It was observed in PICON that the above described event can happen in several different ways particularly when more than two aircraft are involved. Because of the pairing rule, for example, a third aircraft can be heading for the case $t_{c3} \leq T_{cm}$ but will not enter into calculation until $t_{c3} < t_{c2}$ at which time t_{c3} can be $\leq T_{cm}$. (Even if t_{c3} is not $\leq T_{cm}$ it may be destined to become so, because of the non-zero radius of turn in avoidance).

A solution to the above problem lies in forming a θ_{nc} that is not a function of t_{cm} . Not wishing to expand the list of variables and the size of the program, the three variables already defined were used to partially solve the problem as described below.

If $0 < t_{mi} < t_{ci}$, the reference aircraft will be nearest the i th aircraft at some time in the future that is less than t_{ci} ; but t_{ci} is the time that must elapse before a collision can occur. Therefore the nearest passage must not be a collision. Furthermore, the resultant θ_{nc} is not a function of t_{cm} , and so problems due to quantization error in conflict resolution have been eliminated. The new set of conditions for resolution of a conflict is given below:

$$1) \quad t_c > T_c$$

$$\text{OR} \quad 2) \quad t_m < at_c \quad \text{where} \quad 0 \leq a < 1$$

$$\text{OR} \quad 3) \quad t_{cm} > T_{cm}$$

In PICON, "a" was set to 1/2 and it was observed that the quantization problem disappeared. Unfortunately, there was insufficient time to study the exact effects of the insertion of the new condition. An algebraic analysis is given in Appendix B which shows that there is not always a solution for θ_{nc} since t_m is only partly determined by the reference aircraft.

B. En route and Approach Control

It is generally conceded in this work that the ability to predict air traffic positions deteriorates as the time interval of prediction increases; yet the ability to predict air traffic behavior is one of the most important factors in the design of control logic. Ideally, a designer would like to make accurate predictions covering time intervals of hours in en route control. It was decided that worst case errors in predicted positions in real traffic would be trivial if the time interval of prediction never exceeded 15 minutes.

For this reason, 15 minutes was chosen for PICON en route control even though the Cornfield System guarantees zero error in predicted positions.

The short time interval used in en route control has the effect of steering a typical en route flight to its destination in 15 minute segments. This is not objectionable in itself, but it is inadequate after imposition of the safety condition. Conflicts must be resolved with no regard for best paths to destinations; similarly, paths to destinations must be chosen with no regard for conflicts or congestion lying beyond the current 15 minute segment of flight. The overall effect is to maintain safety at the expense of optimization. Optimization is discussed separately later in this section. In approach and terminal areas, the prediction time interval need not be as long since a typical flight spends a relatively short time in each area. In PICON, the time interval chosen for approach area is 5 minutes and that chosen for terminal area is 3 seconds. The latter time interval was chosen because of the relatively high degree of precision required in terminal control.

1. Shell

The shell is a small section of the program whose purpose is to isolate the major functions, such as terminal control, en route control, etc. It is described briefly here to help fully orient the reader. A more detailed flow chart of the shell is given in Fig. 4.

Normally, program control is in the shell in an "idle" state. In this condition, ILLIAC is in a loop, reading data, all of which is discarded except a clock signal. (The basic time interval used in PICON is 3 seconds and this time interval is, of course, marked by a clock signal.) Upon receipt of a clock signal, the shell will modify itself so as to start reading a series of frames. A frame consists of all of the targets in the system and all frames

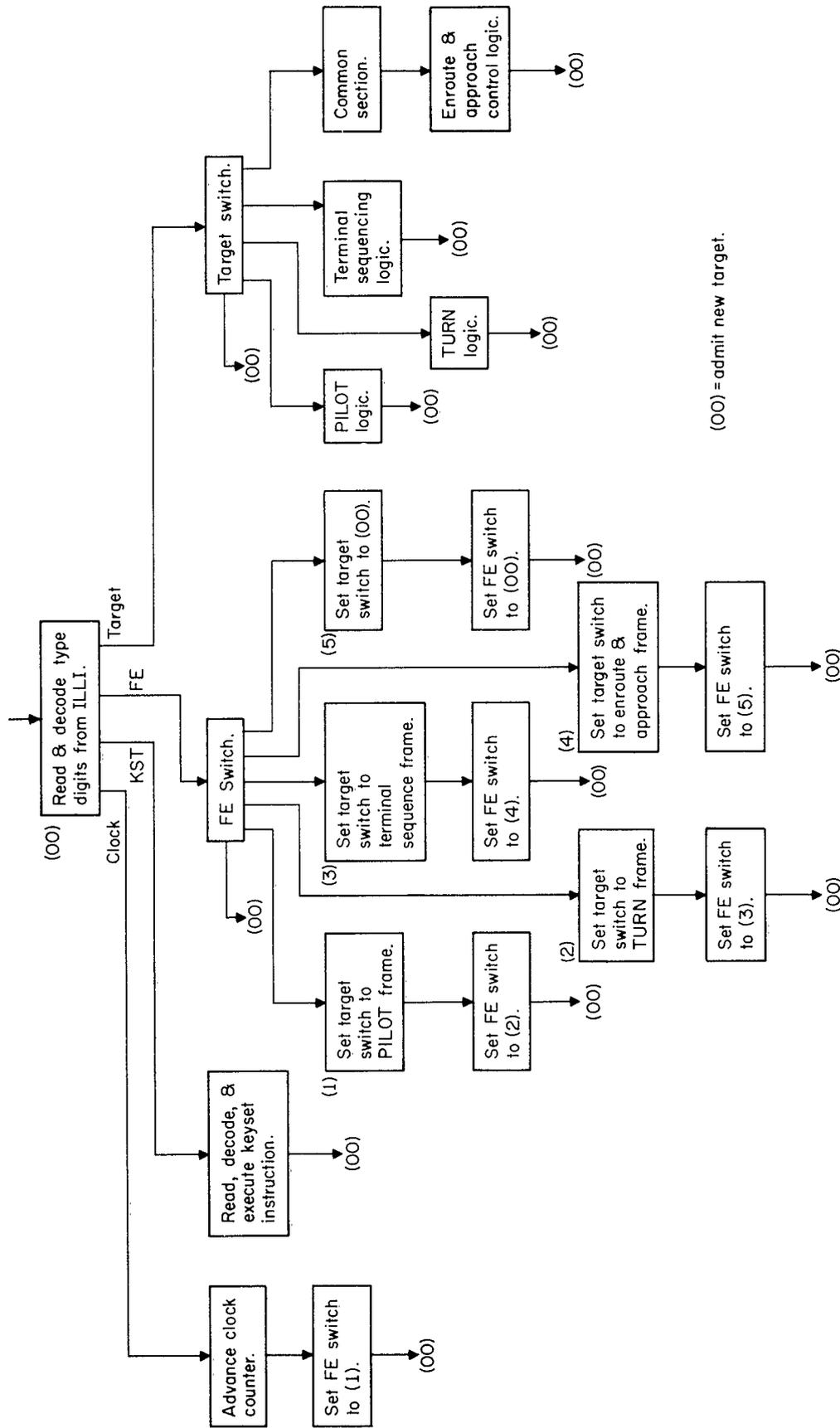


Fig. 4. Shell

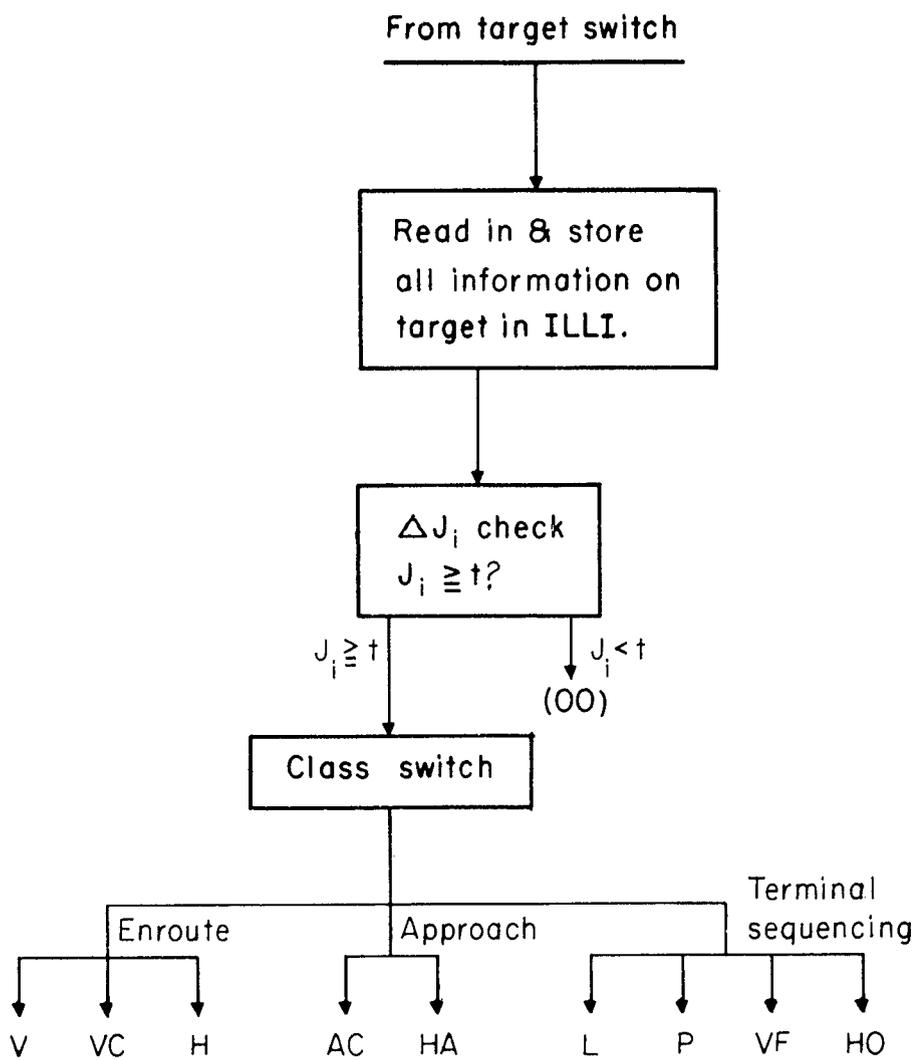
are marked by a signal from TASC called frame end. The shell is programmed so that frames will be used as follows:

frame 1	PILOT
frame 2	PILOT (execute all turns)
frame 3	terminal sequencing
frame 4	en route and approach control
frame 5	collision sort (if necessary)

All control decisions are made in frames 3 and 4; frame 5 may or may not occur depending upon whether or not the sorter was entered in frame 4. After frame 4 (or 5 if there was a sort) the shell is returned to the idle state in anticipation of the next clock signal and repetition of the entire cycle. The functions executed in frames 4 and 5 can be repeated in frames 6 and 7, 8 and 9, etc. until every aircraft requiring a collision sort has been sorted. This repeating process is extremely time consuming and was influential in establishing the maximum number of controlled aircraft in the simulation. The following discussion is a description of the events occurring in frame 4.

Common Section. In frame 4, each target is read into ILLIAC and processed first in the common section (Fig. 5). The two functions executed in common section are ΔJ_i check and class switching. If a target passes the ΔJ_i test ($J_i \leq t_0$), ILLIAC control is transferred to the proper set of logic via the class switch. Classes are allocated as indicated below:

V	vector	en route
VC	vector conflict	en route
H	hold	en route
AC	approach control	approach control
HA	hold approach	approach control



(OO) = Admit new target
for control processing

Fig. 5. Common Section

HO	terminal hold	terminal sequencing
VF	vector final	terminal sequencing
P	maneuver to localizer	terminal sequencing
L	landing	terminal sequencing

2. Conflict Overtake

Conflict overrule is a tiny set of logic used in control. Its purpose is to overrule conflicts with factors that are not easily obtained in the collision sort. In the later versions of PICON, the only factors used in overruling were those pertaining to the arrival of the reference aircraft. An example of overruling is given below.

If the collision sorter declares a reference aircraft in conflict, then the reference aircraft has

$$t_c \leq T_c$$

and $t_m > at_c$

and $t_{cm} \leq T_{cm}$.

The conflict should be overruled if

$$t_m > t_{aa} + T^*$$

where t_{aa} is the time interval necessary to reach the perimeter of the approach area and T^* is an arbitrary constant chosen in PICON to be 2 minutes. The expression simply says that the reference aircraft will be an arrival at least 2 minutes before its nearest passage occurs; since it is known that approach logic can handle the conflict in a safe and expeditious way, the conflict should be overruled.

The constant T^* , as used above has the effect of creating a buffer zone T^* minutes wide on the inside of the approach area boundary. Any conflict falling in the band is to be resolved by en route logic and any conflict falling in approach area but not in the band shall be resolved by approach logic.

3. En route Logic

En route control logic is described briefly in the following paragraphs, and in detail in the following pages. The only traffic that can enter into conflict is that which meets the proximity condition, i.e. $t_c \leq 15$ minutes.

In the complete absence of other traffic, an en route flight will be vectored directly to its destination. It will be processed by automatic control each time it has traversed half the distance remaining to its destination.

In the presence of other traffic, an en route flight will be processed more often than described above, depending on proximity of other flights. If a conflict is found, the en route flight will be turned in the direction requiring the shortest turn. Thereafter, control will check the heading to destination for conflict; if there is no conflict, the flight will be turned back to a heading to its destination; if there is a conflict, present heading will be checked and either maintained or modified depending on whether or not there is conflict in present heading.

If ever an en route flight is turned more than 90° from its heading to destination, it becomes a candidate for en route hold. The candidate is either held or maintained on its outbound heading, depending on whether it is safe to hold. If it is safe to hold, the flight will be held until its heading

to destination is free of conflict within the proximity area (proximity condition is $t_c \leq 20$ minutes for exit from hold). Each en route-holding flight is processed once every minute.

Three classes have been defined for en route logic: 1) vector class (V), 2) vector conflict class (VC), and 3) hold en route (H). Flights in class V are not in conflict; navigation is the primary objective. Flights in class VC are in conflict, and safety is the primary objective. Flights in class H are executing a standard 1 minute holding pattern in en route air space; navigation is the primary objective. The logic of each of these three classes is given in detail below. Logical flow diagrams are depicted in Figs. 6, 7, and 8.

Vector Class, Fig. 6. The following paragraphs consist of expansion and justification of the logic of class V, depicted in Fig. 6. Underline is used to indicate correspondence with the flow chart.

In the logic of class V, the first decision made is in the arrival test. Since approach area is circular, 30 miles in radius and centered on the outer marker, the test is for distance from the outer marker. If the flight is less than 30 miles out, it is an arrival. It should be noted that an arrival does not get collision sorted, however safety is assured as follows: the last time the arrival was sorted it must have been "not in conflict," otherwise it would not be in class V; but the last time the arrival was sorted was only 3 or 6 seconds ago since ΔJ_1 was set to the smaller of $k_2 t_{aa}$ or $k_1 t_c$ (the smallest unit of time is 3 seconds). Therefore, the largest possible violation of proximity is 3 seconds, which is safe.

If the flight is an arrival, en route logic does the routine functions of selecting the nearest entry point and transmitting a heading such that the flight will be vectored toward the entry point. Entry points are points of entry to

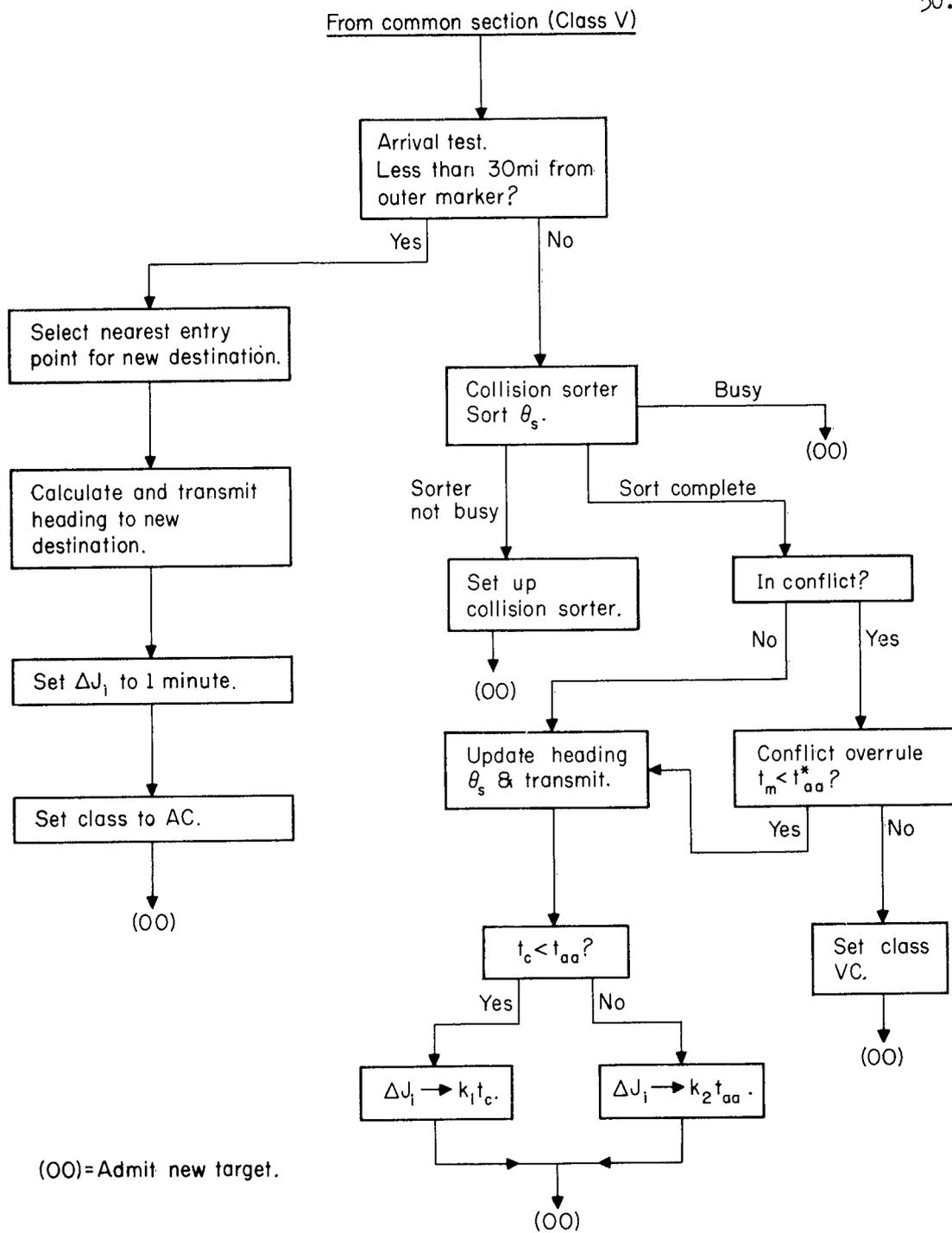


Fig. 6. En route Class V.

terminal area and are described in detail under Terminal Control; in PICON, there are 4 entry points, each about 5 miles from the outer marker.

The next step in "arrival logic" is to set ΔJ_1 to 1 minute. The purpose of this is to avoid an unnecessary hold at the perimeter of the approach area. High speed traffic crossing the perimeter of approach area will almost always find a conflict ahead because of its high speed. The reason such a conflict is not found while the flight is in en route area is that it would have been overruled; upon crossing the perimeter, however, there is an abrupt change in the overrule condition (t_{aa} is zero for a flight at the perimeter but t_{ep} is d_{ep}/s_i after passing the perimeter). If the arrival observes the approach-area speed limit (318 mph), it will be at approach speed in less than 1 minute, at which speed there usually is no approach-area conflict and holding at the edge is eliminated. Safety is assured as follows: the only class that can be an arrival is the V class, but this class cannot be in conflict. Therefore, no-conflict conditions prevail as the flight crosses the perimeter of approach area and in this case, these are:

$$t_c > T_{cm}$$

or $t_m > T^*$ since $t_{aa} = 0$.

In PICON, T_{cm} and T^* are both set to 2 minutes. A collision cannot possibly occur in less than 2 minutes if the arrival continues at normal cruise, and only then if the arrival and an ith aircraft turn to head-on collision courses. If the arrival slows down, there is even more time, of course. Therefore, setting ΔJ_1 to 1 minute to allow for speed reduction, leaves approach control with at least 1 minute to detect and correct a worst case conflict; safety is thus assured for the arrival.

The arrival must then be "handed-off" to approach control and this is done, in PICON, by setting the arrival to class AC. Thereafter, the arrival will be processed by the logic of class AC.

If the flight is not an arrival, it will be processed by en route logic, the first step of which is to enter the collision sorter. The sorter is instructed to sort with θ_s (θ_s is heading to station or destination; θ_p is present heading). The reasoning in choice of θ_s is as follows: It is always desired to turn an en route flight to a heading to destination. If the heading to destination is not in conflict, the flight will be turned to that heading; if the heading to destination is in conflict, the flight will be set to class VC, where a sort using θ_p will follow immediately. If, in VC logic, it is found that there is no conflict on θ_p , the flight will be set back to class V with an appropriate ΔJ_1 . Therefore, it is safe to sort with θ_s , at this point in class V logic, with no regard for θ_p .

The three exits from the collision sorter have already been described. If the sorter was not busy, the flight currently being processed will be selected as reference aircraft and it is then necessary to set up the collision sorter. Setting up the sorter consists mainly of two things: 1) setting the proximity and future proximity parameters to the proper value, and 2) setting the filter to "select and reject" classes appropriate for the reference aircraft. It is necessary to set the parameters because all classes do not use the same parameters (more detail in en route hold logic). Setting the filter has been described earlier. In PICON, the filter is set so that in route class V logic is sorted as follows:

<u>Admit for sorting</u>	<u>Reject</u>
V vector	HO terminal hold
VC vector conflict	VF final vector
H en route hold	P maneuver
AC approach class	L landing
HA approach area hold	

The only remaining exit from the collision sorter occurs after the sort is completed, at which time the collision-sorter storage bank contains all information pertaining to the safety of the reference aircraft. Exit is directly to the conflict test which is the simple query "is the reference aircraft in conflict?" If the answer is yes, the conflict is subject to overrule in conflict overrule (described earlier). If the conflict is not overruled, the flight is set to class VC for immediate conflict resolution. Note that ΔJ_1 is not set, thereby insuring immediate processing in class VC.

If the reference aircraft is not in conflict, then control processing consists of routine matters of up-dating and transmitting the heading to destination. The choice of the smaller of $k_1 t_c$ or $k_2 t_{aa}$ is determined by testing for the smaller of t_c or t_{aa} . The effects of this choice have been described in detail earlier.

Vector Conflict Class, Fig. 7. Any flight in VC class is in conflict and the purpose of the logic is to eliminate the conflict. As in the previous section, underline is used to indicate correspondence with logical blocks of Fig. 7.

Since the present heading (θ_p) is the primary concern in conflict resolution, the collision sorter is instructed to use θ_p in the sort. After accepting the current flight as the reference aircraft the collision sorter is set up as in class V. Exit after the sort is to the conflict test, also

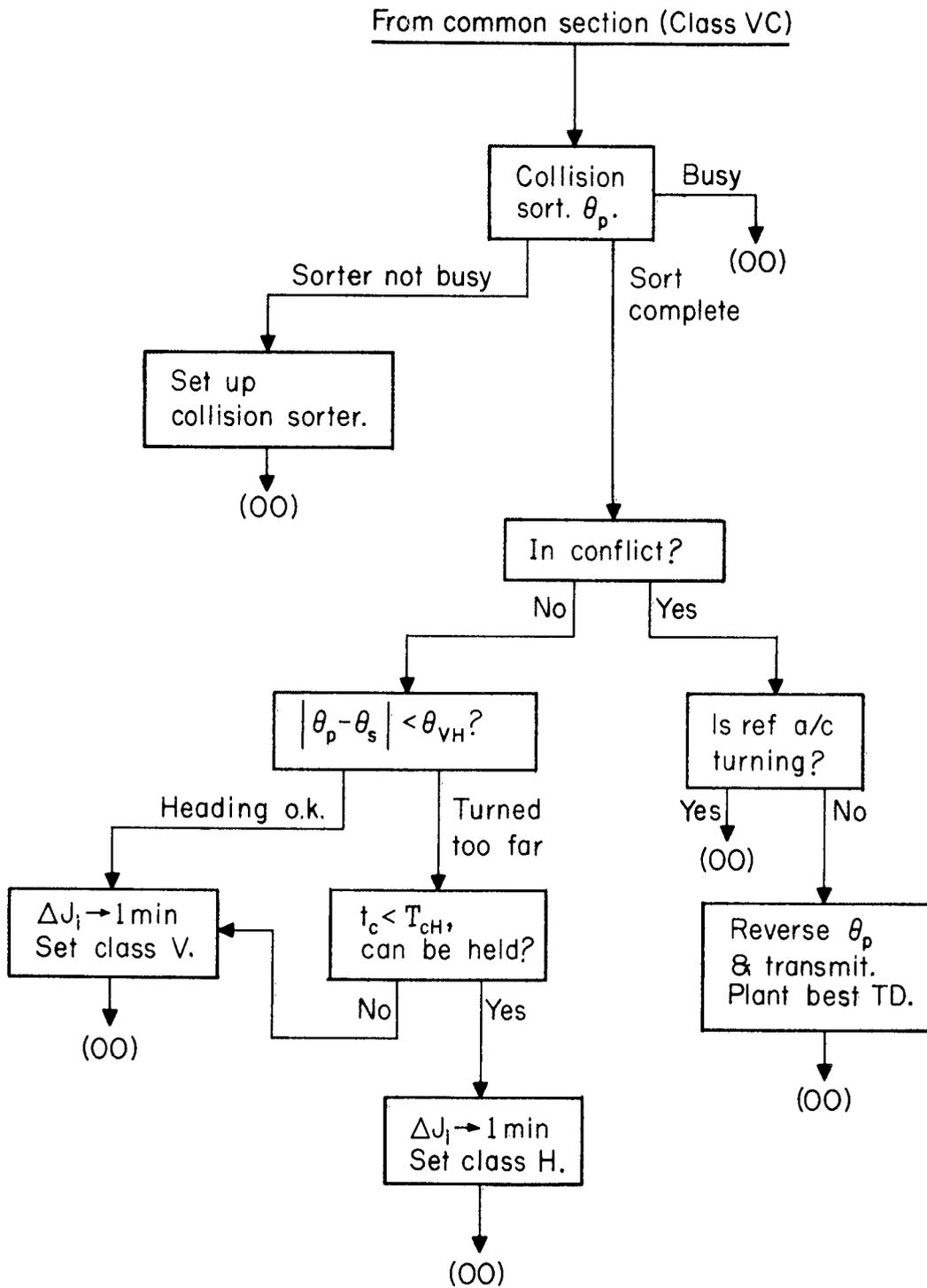


Fig. 7. Logic of Class VC

similar to the logic of class V. Conflict overrule is omitted as it would be unnecessary repetition of what just happened in class V logic.

If the answer to the query "in conflict?" is yes, the logic of class VC must be such as to eliminate the conflict. The logic has only one ability in the "one altitude" model and that is, turn the reference aircraft. It has a choice of turn direction, however, but cannot be allowed to reverse the choice of turn direction, once made, until the conflict is eliminated. (If turn direction for each conflict is not immutable, a hopeless form of vacillation can occur in conflict resolution). In PICON, the turn direction is always that requiring the shortest turn. Referring to Fig. 7, it is seen that the choice of turn direction can only be made if the flight in conflict is not turning, thereby meeting the condition referred to above. Reversing θ_p is an expedient way of starting the iterative process described earlier.

It is conceivable, under conditions of high congestion, that a flight may have its heading changed so that the new heading lies in the reverse half of the compass i.e. the flight is actually flying away from its destination. This is undesirable except perhaps under some truly extreme condition; it is preferable to hold a flight rather than allow such dilatory behavior. The test $|\theta_p - \theta_s| < \theta_{VH}$ is intended to detect unintended departures from the area. In PICON, θ_{VH} is 90° . If a target heads away from destination in conflict resolution it is set to class H (en route hold) provided the holding pattern is safe. Safety at this point is determined by proximity of other aircraft which can be measured by means of t_c . If $t_c < T_H$, hold is not safe ($T_H = 4$ min in PICON) and since there is no conflict, the reverse heading is accepted and the flight is set to class V with a ΔJ_1 of 1 minute. These two steps subject the flight to review under class V logic in 1 minute. Class V

logic will return to this logic (VC) if the same conditions prevail, and the flight will be again tested for holding.

En route Hold Class, Fig. 8. The main objective of hold logic is to get the flight off hold and on its way as soon as possible. Underlinings are used to indicate correspondence with the logical diagram of Fig. 8.

As indicated earlier, a holding flight will be operating at reduced speed. It is necessary to anticipate the increase in speed in the transition from hold to cruise. This is done by planting the cruising speed of the flight in the collision sorter prior to entry. Since the only routes considered in PICON are those leading directly to destination, the collision sorter is instructed to sort with θ_s .

Holding flights in PICON get processed once every minute. After the first minute on hold, a flight will have completed a 180° turn that may be quite wide because the turn was entered at cruising speed. The shift in position due to the turn frequently eliminates the original conflict with the result that the flight is set back to class V and turned toward destination, only to find the same conflict again. Since this is a problem due to approximation of holding pattern position, it was not studied in detail and a crude solution for the problem was devised. The solution in effect makes it more difficult to get off hold than it is to get on hold. This effect is brought about by setting T_{cm} to T_{cmH} where $T_{cmH} > T_{cm}$. Thus a flight that has been set to hold had $t_{cm} < T_{cm}$ (2 minutes) but will not be released from hold unless $t_{cm} > T_{cmH}$ (4 minutes). A similar effect was observed in proximity and it was necessary to set T_c to T_{cH} where $T_{cH} > T_c$. In PICON, $T_c = 15$ minutes and $T_{cH} = 20$ minutes.

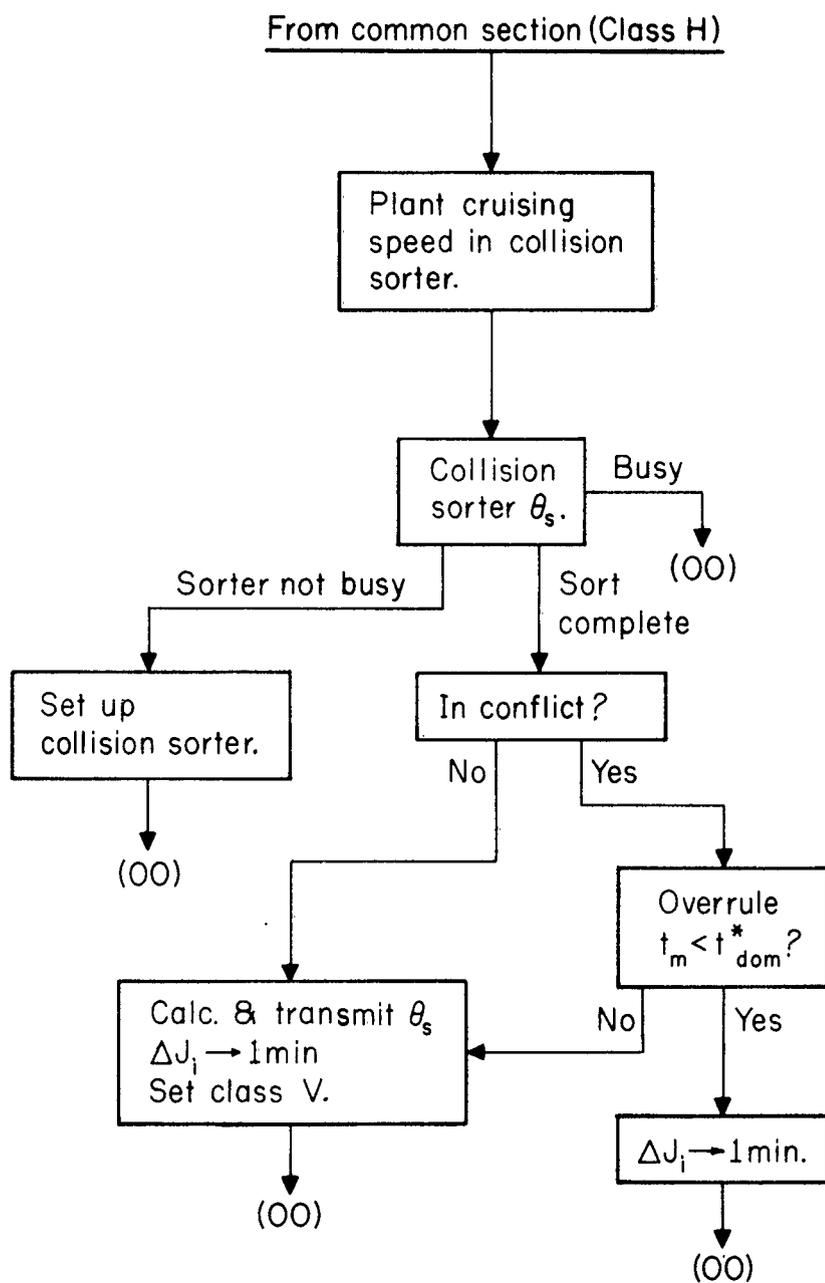


Fig. 8. Logic of Class H.

The above solution introduced a new problem. It is possible for two en route holding flights to capture each other, i.e. neither can get off hold because of the presence of the other. This problem was solved by rejecting holding flights from the collision sort when the reference aircraft is holding. Sorting of H class aircraft is filtered as follows:

<u>Admit for Sorting</u>	<u>Reject</u>
V (vector)	HA (approach area hold)
VC (vector conflict)	HO (terminal hold)
AC (approach control)	H (en route hold)
	VF (final vector)
	P (maneuver)
	L (landing)

All other functions depicted in Fig. 8 are similar to those described earlier and so will not be repeated.

4. Approach Control Logic

Approach area is defined as the area between the perimeter of the approach area circle (30 miles radius with center at the outer marker) and the terminal-sequencing area. Terminal-sequencing area is roughly the area enclosed by the four entry points each of which is approximately 5 miles from the outer marker (see Fig. 11). As described earlier, all traffic will enter the approach area well dispersed and with nearest entry point already selected and assigned. Approach logic is described briefly in the following paragraphs and in detail in the following pages.

The four entry points, along with the rules described below, have the effect of creating 4 non-overlapping sectors in the approach area. Each sector is pie shaped with its apex at the entry point. Traffic in each sector cannot interfere with traffic in other sectors, and there can be no outbound flights in any sector.

Any flight crossing the approach-area boundary is not processed for the first minute. Safety is assured during this interval as described earlier. Thereafter, control will steer the flight toward its entry point. In the absence of other traffic in the sector, the flight will be processed each time it has traversed half the distance remaining to the entry point. Upon reaching the entry point, the flight will be held until selected for terminal sequencing by terminal control.

If there is other traffic in the sector, the approaching flight will be processed more often than described above, depending on proximity. If a conflict is found, the most distant (from the entry point) of the two flights will be held and thereafter processed once every minute. A conflict cannot occur unless the proximity condition for approach area is met ($t_c \leq 5$ minutes). A flight that is holding, will remain holding until there is "no proximity" in which case it will be vectored toward its entry point.

Two classes have been defined as approach control classes: 1) approach control (AC) and 2) hold approach (HA). Traffic in AC class will be steered directly to respective entry points; if a conflict occurs, the most distant of the two conflicting flights will be held. Traffic in HA class will be held on a radial emanating from the entry point; as soon as a holding flight is free of conflict, it will be set to class AC.

Approach Control Class, Fig. 9. As done earlier, underline is used to indicate correspondence with the flow chart of Figure 9.

The flow chart of Figure 9 contains very little that has not already been discussed. Arrival at the entry point is declared if the distance from the entry point is less than 1 mile. A flight reaching its entry point is set to class HO which is the terminal-area-holding class. All HO class flights execute a standard 1 minute holding pattern in which the inbound

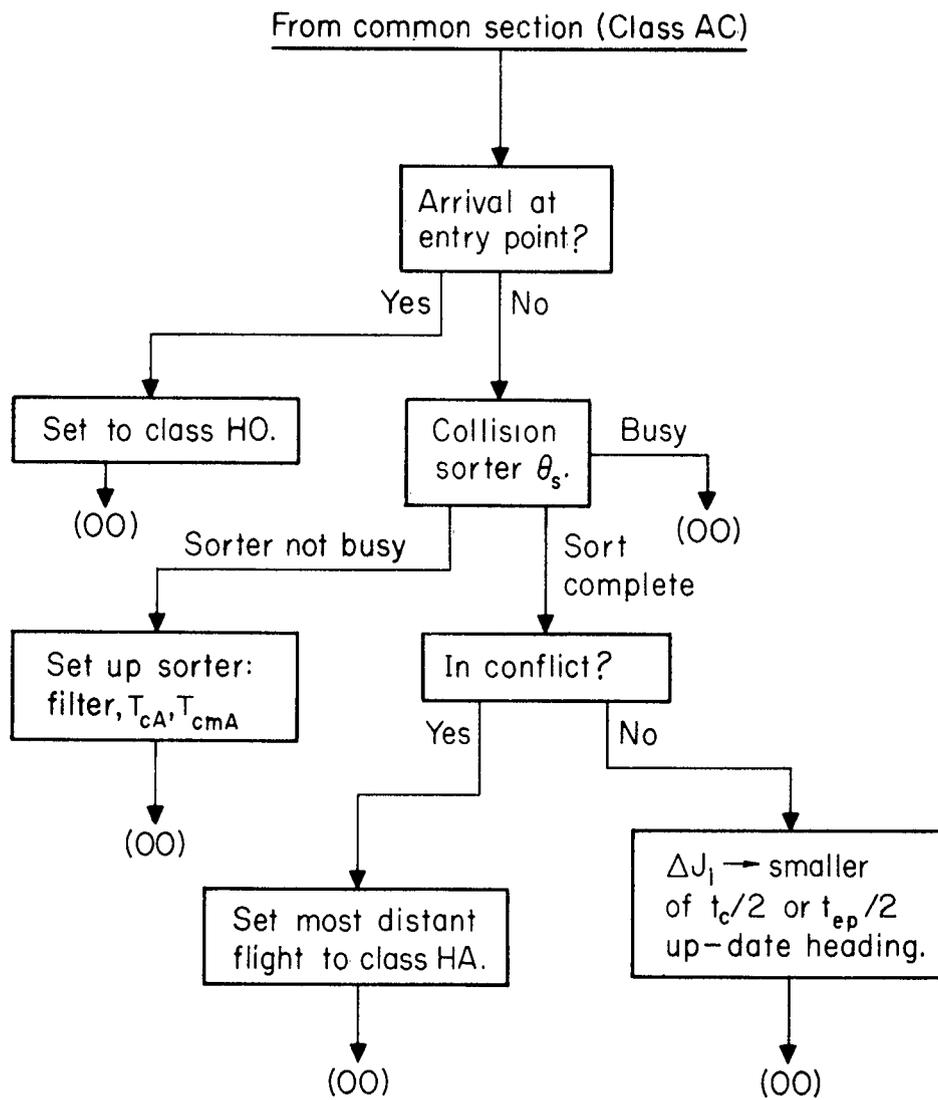


Fig. 9. Logic of Class AC.

leg is a radial with respect to the outer marker. There may be up to four HO class targets simultaneously awaiting terminal sequencing. On the contrary, if the terminal area is vacant, a flight will only remain in HO class for 3 seconds.

A flight that is not an arrival is collision sorted and sorter set up consists of setting the filter and the spacing parameters. The filter is set to sort as follows:

<u>Sort Against</u>	<u>Reject</u>
AC approach	en route classes V, VC, H
HA hold approach	VF final vector
HO hold entry point	L landing.

In addition, the filter is set to reject all targets in AC, HA, or HO that are not in the same sector as the reference flight. The proximity parameter is set to T_{cA} (5 minutes) and the future proximity parameter is set to T_{cmA} (3 minutes). Since it is known that the only evasive maneuver to be executed in approach area is hold, it is only necessary to choose these time intervals so as to guarantee sufficient "time-space" to hold at any time.

Having changed the proximity parameters at entry, the collision sorter will declare a conflict if $t_c \leq 5$ min. and $t_{cm} \leq 3$ min. This is determined in the conflict test. If a collision lies ahead, the most distant of the two flights involved will be set to class HA. If the two flights are equidistant, the choice is arbitrary. If there is no conflict, the heading to entry point is up-dated and ΔJ_i is set to the smaller of $t_c/2$ or $t_{ep}/2$ with the same reasoning as described in Class V.

Hold-Approach Class, Fig. 10. Traffic that is holding in approach area is sorted using θ_s where θ_s , in this case, is the heading to the entry point.

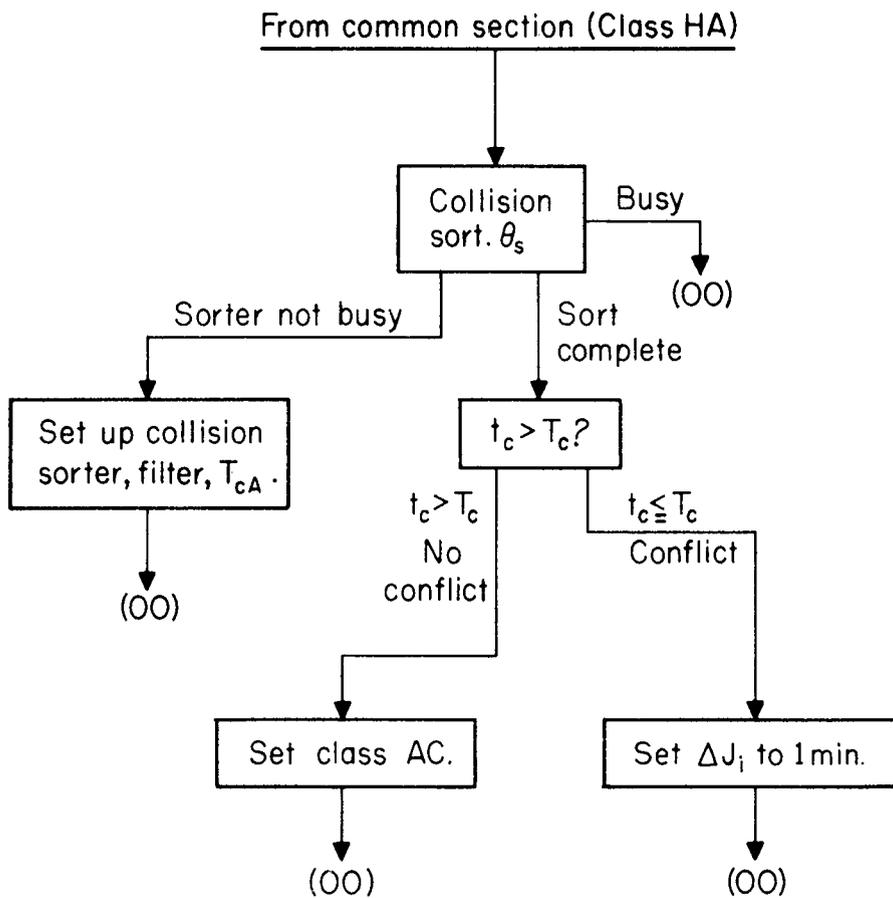


Fig. 10. Logic of Class HA.

If the sorter is entered at a time during which it is not busy, the first exit will be to set up the collision sorter. The filter is set to "sort and reject" as listed in class AC except that holding aircraft are rejected for the same reasons given earlier. Parameters are chosen so that hysteresis is generated i.e. $T_{CHA} > T_{CA}$, which in effect makes it more difficult to exit from hold than to enter hold. At the completion of the sort the conflict decision is simply the no-proximity condition ($t_c > T_{CHA}$?). This decision does not reflect future proximity in any way, but it is noted that if there is no conflict, the flight is released from hold and set to class AC without setting ΔJ_1 . This guarantees that an AC sort will follow immediately in which future proximity is determined; if the AC sort reveals conflict in future proximity, the flight will be set back to hold with ΔJ_1 of 1 minute. In this case, the flight will have spent 3 seconds in class AC

If the holding flight is found to be in proximity (and therefore in conflict) with another flight, it will simply be retained holding and ΔJ_1 set to 1 minute. Thus a holding flight will be processed once every minute until conflicts have disappeared.

5. Uncontrolled Traffic (VFR)

The logic of uncontrolled traffic is relatively simple in terms of the tools and techniques described thus far. Several different versions have been used in PICON from time to time but ultimately had to be removed to make room for more important functions. An uncontrolled flight can be a reference aircraft for collision sorting just like any other flight and can also be treated with ΔJ_1 logic. In sorting, the collision-sort filter is set to reject all uncontrolled traffic thereby limiting the sort to conflicts capable of being

resolved. If a conflict is found, it is only necessary to set the conflicting controlled flight to class VC.

Clearly, this type of logic subjects controlled traffic to the whims of uncontrolled traffic, particularly in a "one-altitude" model. Since the safety condition for controlled traffic must be met, there is no way to avoid this problem entirely. In a real system, the magnitude of the problem could be reduced to trivial proportions by equipping all traffic including uncontrolled, with automatic altitude transponders. Another solution is to require cross-country uncontrolled traffic to identify itself (verbally) with a statement of cruising altitude. The latter method is no problem in the system described in Part I of this report. Still a third solution lies in the fact that en route air space will undoubtedly have a floor determined by radar or VHF coverage. Such a floor forms a natural boundary for restricted air space, and can be used to maintain altitude separation between controlled and uncontrolled traffic.

6. Optimization Logic

Though no work has been done on a set of optimization logic, some of the properties that it must have, can be specified from our work with PICON. Optimization can be regarded as long-range anticipation in contrast to collision avoidance which is short-range anticipation.

The first question of concern is whether or not it is adequate to optimize traffic flow within each control area while ignoring traffic in all other control areas. The question is important because a negative answer calls for extensive expansion of the automatic communication facilities linking control centers. This is not desired, of course. The question is answered in the next paragraph.

If optimization is not confined to each control area, then routing ability is expanded to cover control areas far removed from the direct line of intended flight. Noting that each control area will cover several states, it follows that it would take an enormous amount of congestion to justify routing over such devious routes. A conservative estimate of such congestion is well over 1000 flights, which is in excess of system capacity. We conclude that the required congestion cannot occur and it is, therefore, adequate to confine optimization to each respective control area.

As mentioned earlier, en route control must involve prediction of air traffic positions over long time intervals. It has also been pointed out that accuracy is lost as the length of the time interval increases. If error is defined as the difference between predicted position and actual position, it is possible to determine an expected error function of future time. This function can probably be calculated but more likely would have to be deduced from studying data on actual air traffic. The function would be used in optimization logic in such a way as to control the percentage of total traffic allowed to enter into conflict.

Since the expected error function is a statistical expression the results would also be statistical. If, for example, every flight in the system behaved exactly as predicted, there would be no conflicts. In contrast, if every flight in the system operated at the extreme of expected behavior then very likely all would eventually enter into conflict. These two cases are limiting cases so that on the average on a typical day, the system would operate somewhere in between. Exactly where it operated would be determined by parameters in the expected error function.

Delay is also affected by choice of parameters in the expected error function. If a "best route" is defined as the route taken when expected error is zero, then delay can be defined as all time in excess of the time required to fly the best route. If the error function parameters are chosen so that on the average there are no conflicts, spacing will be large and delay for each flight will be maximum. Similarly if parameters are chosen at the opposite extreme, all flights that perform with zero error will have no delay; but since most traffic cannot operate with zero error, percentage of traffic in conflict will be high.

Probably the best way to implement optimization would be to lay out routes in straight line segments of some minimum length. This could be done easily by choosing intermediate "destinations" (unknown to all pilots) consisting of coordinates anywhere within the control area. Thus a flight traversing a control area may do so in 4 or 5 straight line segments which may or may not form a single straight line.

In the above paragraphs, it is implied that optimization logic would operate on a "first come first served" basis in which late-comers may well get highly devious routes. This need not be the case, however, as it is reasonable to re-examine previously examined flights and rearrange routings. The effect of using a "revocable routing" philosophy is to spread delay evenly among all flights. This is probably preferable to a system in which late-comers suffer excessive delays. The only limitation in rerouting is the frequency of occurrence for any given flight which, in turn, is limited by pilot tolerance of frequent heading changes.

C. Terminal Sequencing Control

The purpose of terminal sequencing of any form is to maintain the best runway utilization commensurate with safety. For landing traffic, runway utilization can be defined as the number of flights landing per unit of time in the presence of traffic waiting to land.

Runway utilization is limited by the following four factors: 1) accuracy of position information of traffic, 2) equipment failure (if, for example, a radar should fail, spacing must be such that transition to a backup control method will be safe), 3) error due to fluctuations in speed and heading of individual flights and 4) decision-maker overload. In the existing manual system, items 1 and 2 are the major limitations in runway utilization; this is reflected in the currently used sequencing rule which states: only one aircraft shall occupy the air space between the outer marker and the far end of the runway at any one time. The distance from the outer marker to the runway is typically 5 miles.

If position reporting accuracy and reliability are improved, then a new sequencing rule can be devised and the third and fourth limitations listed above become important. Similarly, if airport utilization is increased by increasing the number of runways, then controller overload becomes a limitation. Both factors provide ample motive for devising methods for automatic terminal sequencing. In the automatic terminal sequencing described below, the sequencing rule mentioned above was employed even though it could only result in a 10 or 15 o/o improvement over present runway utilization. The primary goal was to devise methods that could be used with any sequencing rule.

In PICON, terminal area is located in the center of the approach area and contains the runway, the outer marker, the gate, and four entry points deployed as shown in Fig. 11. The control of aircraft movements in the area is designed to bring about safe and efficient landings. The control functions, although merged in implementation, logically may be divided into three parts: assembling the landing sequence, regulating landing separation, and maintaining collision-avoidance restrictions. A typical traffic pattern resulting from terminal area control is shown in Fig. 12.

1. Assembling of the Landing Sequence

When a plane reaches a terminal area entry point, its arrival time at the entry point is recorded and it becomes eligible for insertion in the landing sequence. If the plane is the only one at any of the entry points and no delay is necessary, the plane is immediately assigned a position in the sequence. The assignment of position is irrevocable and is accompanied by the transmission of a heading. (In a real system, a pilot would know that he was in terminal area by observing his distance from the outer marker). If delay is necessary, the plane is directed to fly a prescribed hold pattern at the entry point and is inserted in the landing sequence after sufficient delay accrues.

In general, when two or more planes are waiting at entry points, the landing sequence is formed by adding one plane at a time to the end of the sequence. An addition to the sequence requires two steps: (a) selecting the candidate for assignment to the sequence and (b) assigning the candidate to the sequence at the proper time.

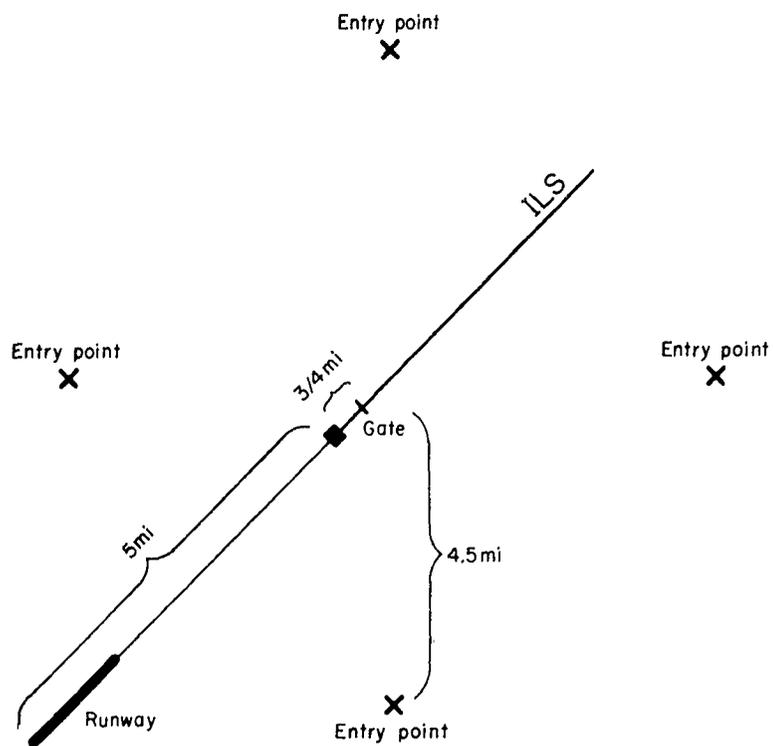


Fig. 11. Terminal Area.

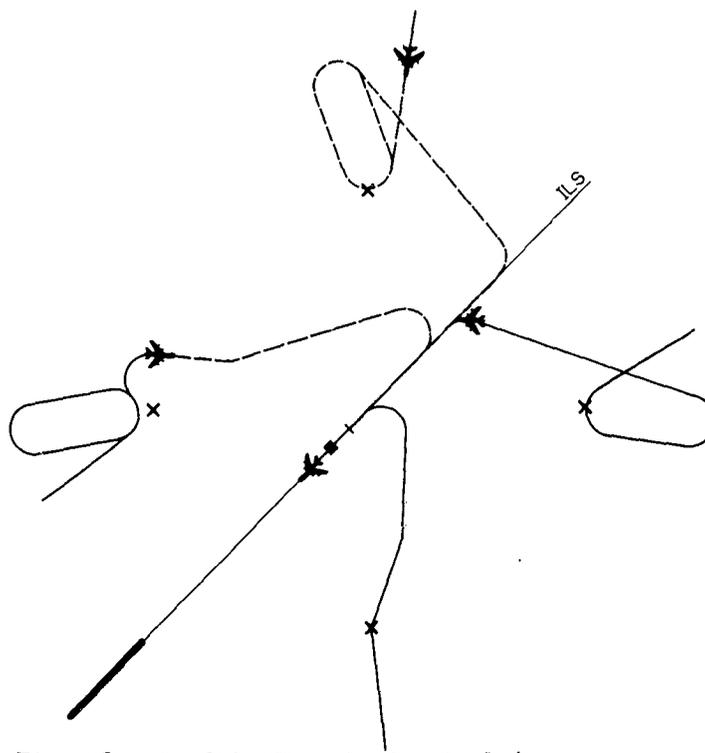


Fig. 12. Traffic Flow in Terminal Area.

Selection of Candidate. The candidate is selected according to the following two rules, in which t^W is the amount of time a plane has waited at an entry point and T^W is a preset parameter. (Superscript notation is used in all of the following; not to be confused with exponents.)

Rule I: If $t^W \leq T^W$ for all planes, select as the candidate the plane with the highest approach speed.

Rule II: If $t^W > T^W$ for at least one plane, select as the candidate the plane with largest t^W , i.e., the plane which has waited at an entry point the longest.

Both rules, when supplemented by the timing considerations stated below, allow the maintenance of maximum landing rate. In addition, Rule I minimizes the total delay for the planes that have reached an entry point¹, while Rule II yields simply the first-come-first-served order. Rule II is designed to prevent the occurrence of long waiting periods which are possible under Rule I. The frequency with which it is used depends, of course, on the setting of the parameter T^W ; in simulation runs with T^W set to 10 minutes, Rule II was rarely invoked.

Assignment to the Sequence. The transfer to the landing sequence is timed so that the candidate, after traversing a short approach path, will land maintaining the prescribed separation. That is, if t_L is the landing time of the last plane in the sequence, t_p the candidate's minimum time required for landing, and \hat{t}_p the time separation prescribed for the candidate, the transfer is made, in principle, when $t_L + \hat{t}_p = t_p$. In practice, prescribed time separation,

1 For proof and a more general discussion of the effect of landing order on runway acceptance rate and total delay see Coordinated Science Laboratory Report R-142.

\hat{t}_p , is increased by a safety buffer \hat{T} , and T_a seconds are allowed for possible errors in time estimates; the transfer is actually executed when

$$t_L + \hat{t}_p + \hat{T} + T_a = t_p.$$

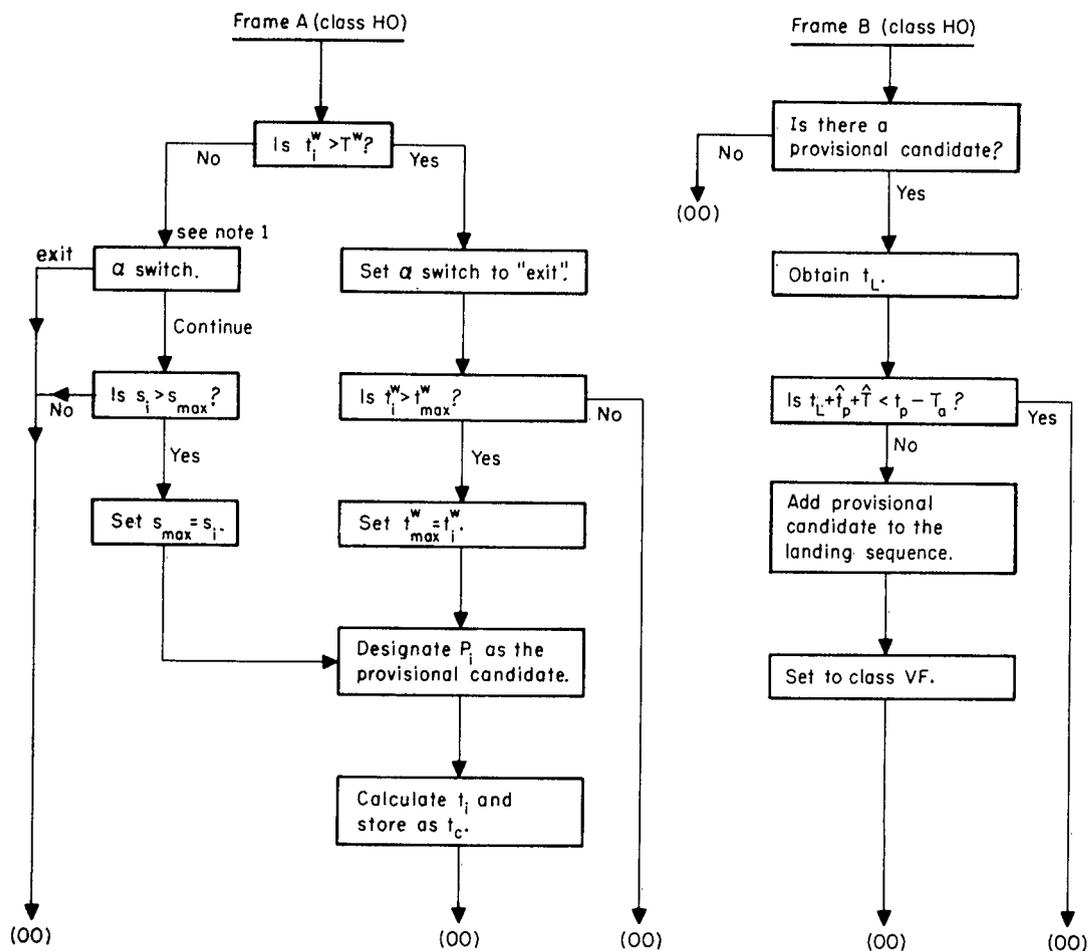
Any necessary delay is then generated by lengthening the approach in the manner described below. Except for missed-approaches, all assignments to the landing sequence are irrevocable.

Programmed Execution. The computer program executes additions to the landing sequence in two frames as shown in Fig. 13. In frame A, the search for a candidate is initiated with Rule I in operation. Each plane's speed is compared with the largest of those already examined and if found to be greater causes the corresponding plane to be designated as the provisional candidate. After all planes have been examined, the existing provisional candidate becomes the candidate. If a plane with $t^W > T^W$ is encountered, speed comparisons are replaced with t^W comparisons and the remainder of the procedure is left unaltered.

In frame B, it is determined whether the proper time for adding the candidate to the sequence has arrived. If not, no action is taken; if yes, the plane is assigned the last place in the landing sequence and steered to final approach.

2. Regulation of Landing Separation

The final approaches are executed along tangential paths whose general pattern can be seen in Fig. 12. Each path consists of the initial regulated part, extending from the vicinity of the entry point to the turn-on arc, and the remaining unregulated part in which the plane first seeks and then follows the ILS (Fig. 14).

NOTES:

1. Switch α is always set to "continue" at the beginning of frame A.
2. (OO) = admit new target.

Fig. 13. Assembling of the Landing Sequence.

Regulated Path. While a plane is traversing the regulated part, its progress is examined every 3 seconds to determine whether a change in landing time is necessary. At each examination, landing time t_i is estimated¹ and compared with t_{i-1} , the landing time of the plane immediately ahead in the sequence. Then if

$$t_i = t_{i-1} + \hat{t}_i + \hat{T},$$

where \hat{t}_i is the required separation and \hat{T} a safety buffer, the plane is scheduled to land at the proper time and, consequently, is left on its present course. If

$$t_i < t_{i-1} + \hat{t}_i + \hat{T},$$

the plane is due to land too early; this information is recorded. If (and only if) the same inequality is obtained 3 seconds later when t_i is calculated again, the course of the plane is changed by $\Delta\theta$ to lengthen the approach path.

The heading correction that will produce the desired change in arrival time can be obtained either directly from a formula and a table loop up, or approximately by iteration of a $\Delta\theta$ increment. In programmed form, the iterative method appears to be substantially more efficient. Its accuracy, of course, depends on the magnitude of $\Delta\theta$ and can be brought to a satisfactory level by taking $\Delta\theta$ equal to the minimum angular resolution of the whole system. The current control program employs the iterative method but, because it was not possible to fit the iteration loop in the available space, the program can execute only one $\Delta\theta$ increment per plane in each 3 second control cycle.

The plane is due to land too late if

$$t_i > t_{i-1} + \hat{t}_i + \hat{T};$$

1 For method of estimation, see Appendix B.

this fact is recorded for use 3 seconds later. If (and only if) the same inequality is obtained 3 seconds later, and if the plane is not already on the shortest path, the path is shortened by a change in heading. Otherwise, no action is taken.

In simulation runs assuming perfect tracking, perfect pilot response, and $\Delta\theta = 8^\circ$, the regulation procedure described above appeared to be satisfactory. First, it produced small landing spacing error. The rule prescribing minimum spacing was that there be at most one plane between the outer marker and the turn-off point on the runway. Since a safety buffer of \hat{T} minutes was used, the actual landing separation that the control program sought to maintain was

$$\text{landing separation} = \hat{t}_1 + \hat{T},$$

where \hat{t}_1 is the time interval required to traverse the distance between the outer marker and the runway turn-off point. Fig. 15 shows the distribution of spacing error for 134 arrivals under overload conditions.

Secondly, the procedure appeared to be effective in keeping down to a reasonable level (about 2 per approach) the heading oscillations which arise from fluctuations in landing time estimates. It seems likely that the oscillations can be reduced to any desired level by changing the rules which govern heading changes. The optimal rules, once established, will require that heading changes occur at least n seconds apart. Two other considerations which also require a guarantee of a minimum time interval between any two successive heading changes are (a) pilot tolerance of frequency of change and (b) loss of reliability in position information obtainable from a tracking computer after a turn. It seems, however, that the minimum time interval

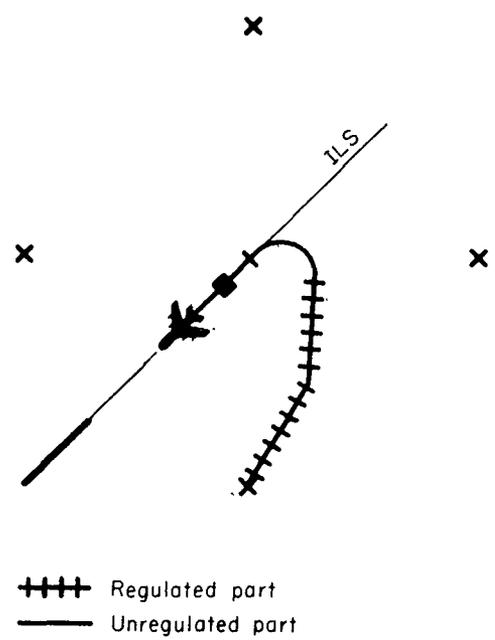


Fig. 14. Approach Path.

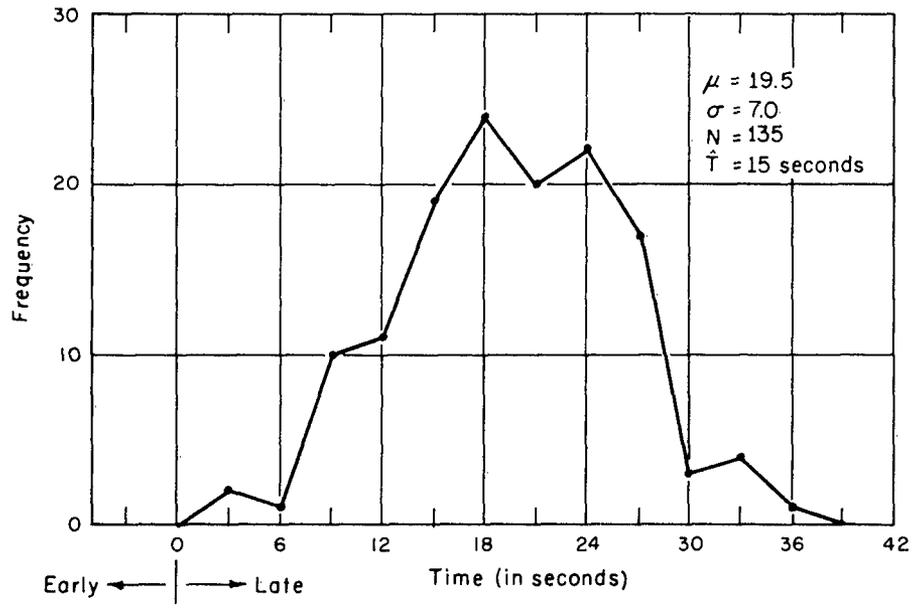


Fig. 15. Arrivals at the Outer Marker.

needed for inhibition of oscillations will be the smallest of the three and that paths free of oscillations can be obtained well within other stability requirements.

Unregulated Path. While a plane is traversing the regulated part of the path, its position is examined every 3 seconds to determine whether the turn-on for aligning with ILS should be initiated. If yes, the plane is given the required heading immediately. The issuance of this heading change marks the point at which control over the plane's heading is relinquished (except for a possible wave-off). The plane is still monitored, however, to provide updated-landing-time estimates and for use in maintaining landing separation for following traffic. When the plane reaches the outer marker and complies with the minimum landing separation rule, it is allowed to continue the glide toward the runway. If, however, it violates the rule, control over the plane is resumed. The plane is given a heading 22.5° away from ILS, and handed over to approach area control for another landing approach via a terminal area entry point.

Programmed Execution. The program controls final approaches in one frame as shown in flow diagrams in Fig. 16 and 17. When a plane just begins the final approach (Fig. 16), it is turned, if necessary, toward ILS at an angle which makes it possible for the plane to align with ILS before reaching the gate. From then on, both position and landing time of the plane are examined every 3 seconds and, when required, turn-on or path-changing directives are issued.

After the turn-on is initiated (Fig. 17) and until the plane reaches ILS, t_1 is updated simply by subtracting 3 from the previous value of t_1 every 3 seconds. When the plane reaches ILS, it is assumed that from here on

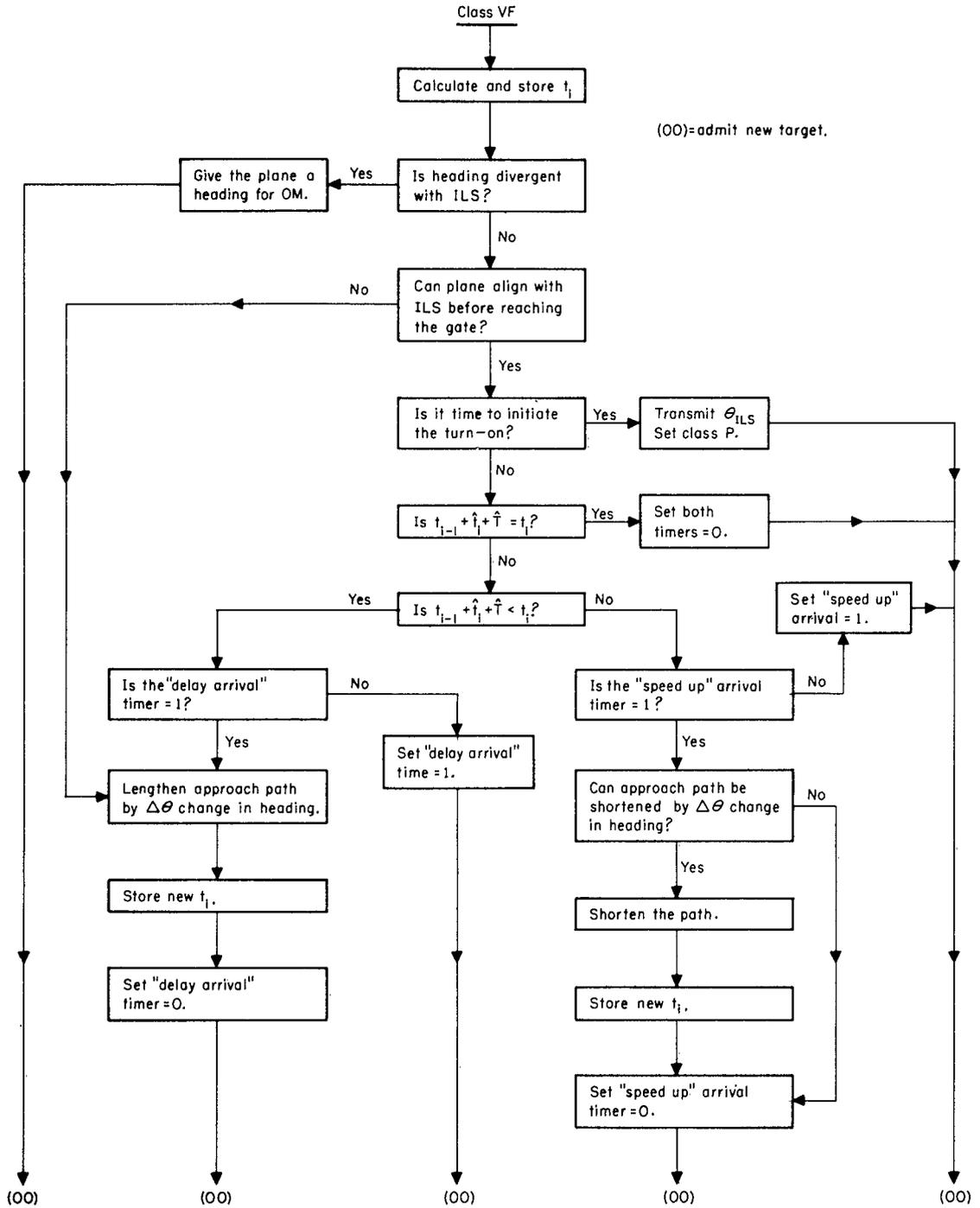


Fig. 16. Regulated Part of Final Approach.

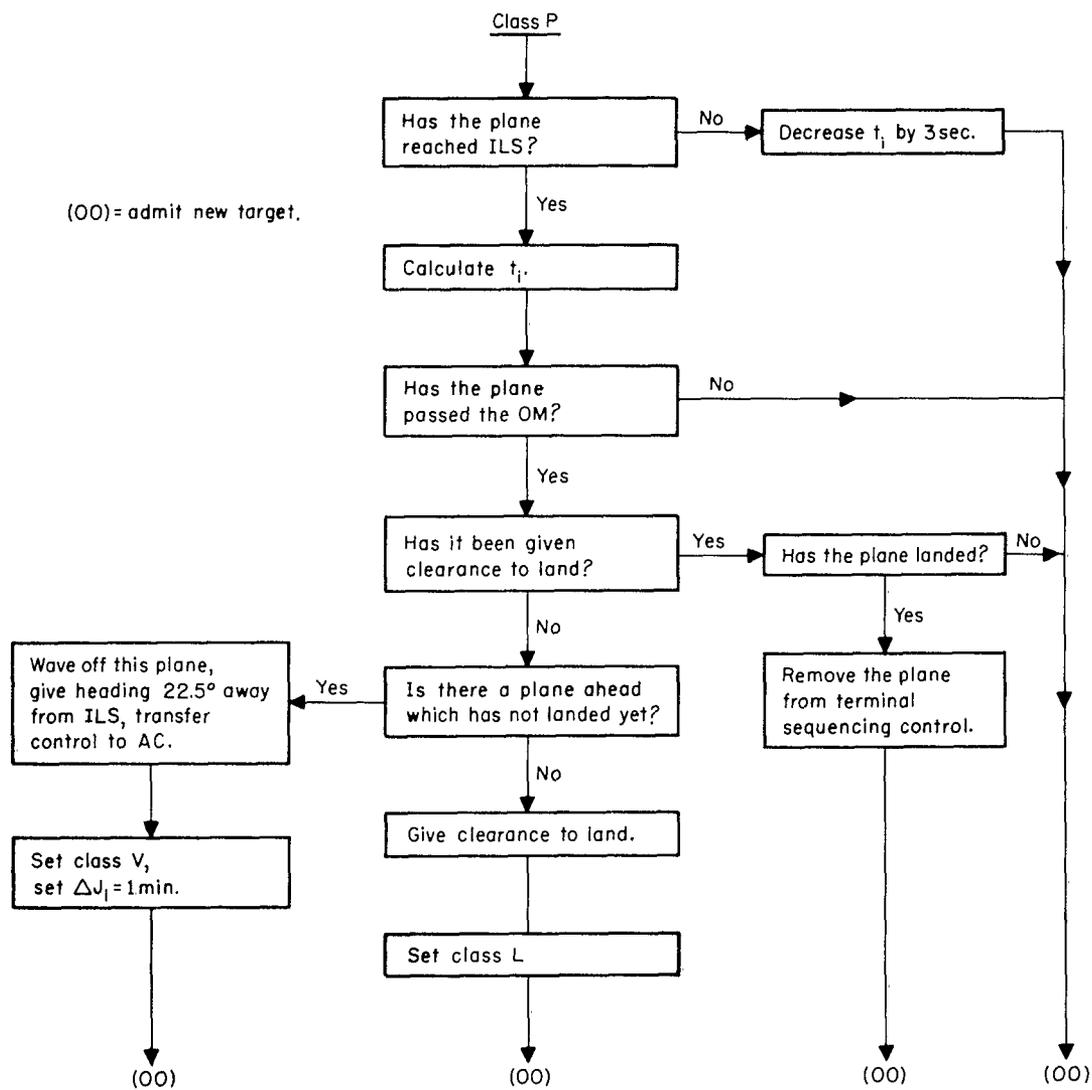


Fig. 17. Unregulated Part of Final Approach.

the plane will follow a straight line and t_1 is estimated on that basis. At the time of the plane's passage over the outer marker, landing separation is examined and a wave-off executed, if necessary. If a wave-off is not required, the plane proceeds to the runway where it is considered to have landed upon reaching the turn-off point. In the current version of the program, wave-offs and landing are determined automatically in order to facilitate simulation runs. In a working system, both decisions could be made by pilots and human controllers and communicated to the control computer through a keyset.

3. Collision Avoidance

During the execution of final approaches and, in fact, throughout the terminal area, collision is prevented entirely by passive means; that is, collisions are prevented by means of restrictions on movements of all aircraft and not through evasive maneuvers chosen after conflict situations arise. More specifically, the technique of passive collision avoidance in terminal sequencing can be described as follows.

Given the following conditions:

1) a physical arrangement of entry points, gate, outer marker, and runway such as that depicted in Fig. 11;

2) $s_i \leq S_{\max}$ where S_{\max} is the maximum approach speed ever encountered;

3) $\frac{S_{\max}}{S_{\min}} \leq K$ where S_{\min} is the minimum approach speed ever encountered and K is a known constant;

4) T , a known maximum delay time to be lost by path stretching in the terminal area;

5) the sequencing rule, i.e. the airspace along the ILS between the outer marker and the far end of the runway shall never contain more than one aircraft at a time.

It can be shown that collisions cannot possibly occur if the above parameters are properly chosen; consequently it is safe for the control computer to execute the sequencing rule with no regard for collisions. By definition, this is passive collision avoidance. In PICON, the physical arrangement of Fig. 11 was used, $S_{\max} = 206$ mph, $K = 1.84$, and $T = .5$ minutes. Factors influencing choice of parameters and a partial proof of impunity are given below.

Entry Points. Entry points must be placed far enough from the ILS to allow adequate path stretching, and sufficiently far from each other to allow safe holding patterns. In PICON, all traffic holding at entry points was held on a radial with respect to the outer marker. This was done out of convenience in programming, since holding pattern orientation is of little consequence in the experimental model. Entry points must be numerous enough to permit maintenance of the maximum landing rate. The only additional factor influencing choice of entry point location is a small effect on the shape of approach boundaries described in the next section.

Normally, traffic entering the terminal area will do so via an entry point. In PICON, however, any terminal holding flight can be started inbound from any point in its holding pattern, in which case, the flight may not cross the entry point. Although this fact complicates the analysis of terminal sequencing, it has been observed to work perfectly well and even, perhaps, enhances the overall performance. In all analysis and diagrams in this report, it has been assumed that all entries to terminal area are via entry points.

Approach Paths. Approach path boundaries are fully specified by the five conditions described above. Applying all of these conditions yields the diagram of Fig. 18 in which the inner boundaries are established by the slowest

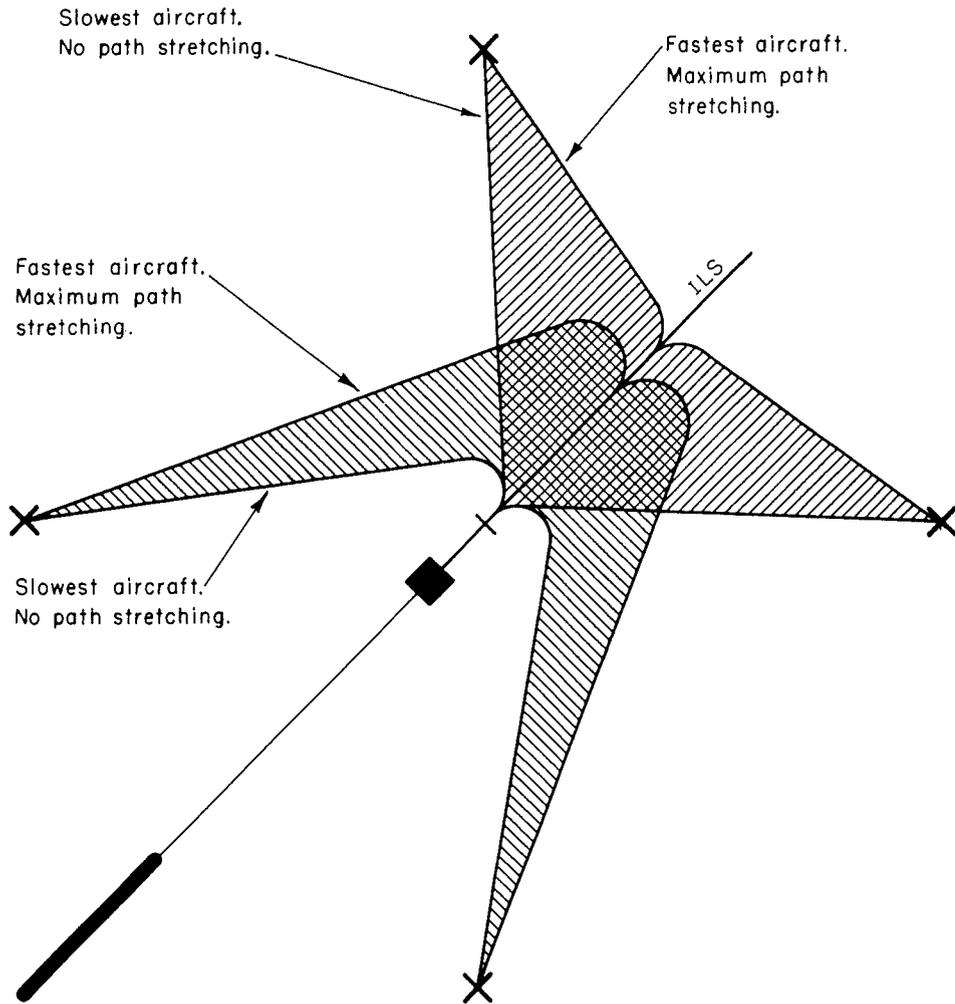


Fig. 18. Approach Boundaries

aircraft with no path stretching ($T = 0$), and the outer boundaries are established by the fastest aircraft under maximum path stretching ($T = .5$ min. in PICON). All traffic that is controlled by terminal sequencing will traverse a path that lies entirely within the shaded area of Fig. 18. The path from entry point to "final turn onto ILS" need not necessarily be straight, as this segment of flight is subject to fine heading adjustments to maintain the sequencing rule. From this diagram (Fig. 18), it is possible to select the worst cases for special analysis regarding collision avoidance. Two worst cases are described below, along with conclusions.

Common-Path Overtake. The two factors aggravating common-path overtake are long common path and widest possible speed differential. From Fig. 18, it is seen that the longest possible common paths are those terminating on the same entry point. These are trivial cases, however, since approach control logic maintains spacings such that two flights cannot possibly cross the same entry point and proceed to an overtake. In fact, the opposite is true; minimum spacing in approach area is such that gaps will occur in the landing sequence if all traffic enters terminal area through the same entry point.

The worst case in common-path overtake is therefore restricted to the "forced" common path, i.e. the ILS, and the worst configuration consists of the fastest aircraft overtaking the slowest. In the following, it is shown that an overtake cannot occur along the ILS in the worst case possible under the five given conditions.

By the sequencing rule, the fastest aircraft must be at the far end of the runway at the same instant that the slowest aircraft is over the outer marker. The problem is to determine the length of common path necessary to include an overtake; this length is d_{ot} , and is measured with respect to the outer marker. Equating the time of flight for each aircraft and solving for d_{ot} ,

$$d_{ot} = \frac{D}{K-1}$$

where D is the distance from the far end of the runway to the outer marker, and K is $\frac{s_{max}}{s_{min}}$. In PICON, D is 4.5 miles and K is 1.84, from which d_{ot} is 5.4 miles.

Referring to Fig. 18, d_b is defined as the distance from the outer marker to the point at which the most distant approach boundary intersects the ILS. Clearly, if

$$d_b < d_{ot},$$

an overtake is impossible in any case that can exist under the five given conditions. By graphical means, it has been found that d_b in PICON is 4.4 miles and therefore, an overtake cannot occur. Furthermore, the nearest passage that can occur in overtake configuration is 1 mile ($d_{ot} - d_b$). This will always occur as the slowest aircraft turns onto the ILS behind the fastest aircraft, with subsequent widening of their spacing.

In PICON, the sequencing rule is actually enforced with a safety time interval, \hat{T} . The purpose of the time interval is to allow for minor variations in traffic behavior. The safety interval operates such that ideally, each landing is followed by a time interval, \hat{T} , during which the air space between the runway and the outer marker is unoccupied. In effect, the time interval extends d_{ot} but since \hat{T} is a "fluctuation period" it cannot be used in the worst case calculations above. Taking \hat{T} into account to determine an average minimum spacing,

$$\text{avg } d_{ot} = \frac{D + \hat{T} s_{max}}{K-1}.$$

Evaluating with PICON parameters, it is seen that the average d_{ot} is 6.5 miles for a \hat{T} of .25 minutes. Since d_b is 4.4 miles, it can be said that on the average, the nearest passage in overtake-configuration will be 2.1 miles.

Common-Area-Crossing Paths. Referring to Fig. 18, it is seen that the common area is clearly defined as the double shaded area; paths can cross only in the common area. The problem for the system designer is to prove that the five given conditions guarantee that it is impossible for two flights to occupy the common area simultaneously. (By symmetry, it is adequate to consider only one side of ILS). Unfortunately, proof of impunity in the common area is quite difficult. A crude graphical proof has been obtained but is not presented here. A more formal proof is being prepared, along with additional analysis of the problems of passive collision avoidance.

4. Departures

PICON was originally designed to include a departure class but it was removed early in the work to make room for more necessary functions. The logic of departure is quite extensive and could actually consist of several classes. Most, or all, of the techniques described in this report can be applied to departure, including passive collision avoidance. Exit points, analogous to entry points, are certainly feasible, as is restricted air space beyond the exit points with altitude separation from approaching traffic. Very likely optimization logic will be influential in controlling departures.

5. Uncontrolled Traffic (VFR)

Clearly the logic of terminal sequencing assumes controlled traffic and since this assumption is not, in reality, the prevailing condition, some comments are in order regarding system operation on a clear day. Since no attempt was made in this work to simulate VFR traffic in terminal or approach areas, the comments are essentially statements of what can be done.

Obviously, the sequencing rule of automatic control provides inadequate runway utilization for operations on a typical clear day. It is also apparent that tower operators at a busy airport do an adequate job of sequencing traffic with good runway utilization. Therefore, it appears that automatic control could be safely terminated at the present airport area boundary (5 miles radius), and the remainder of the flight executed with present day techniques. If a flight chose to maintain automatic control, it could be sequenced as described, with automatic control terminating at the outer marker; the flight would thus be turned over to tower control with ample time remaining for sequencing by tower personnel.

In the terminal area, mixing automatically controlled traffic with uncontrolled leads to difficult problems. The flight choosing to maintain automatic-terminal-sequencing control would very likely suffer excessive delay while uncontrolled traffic occupied the sequencing zone. The only solution to this problem appears to be restriction of airspace, such as reservation of a specific altitude for all controlled-final maneuvering prior to contact with the glide-slope.

IV. EVALUATION AND CONCLUSIONS

Because of numerous factors beyond our control, very little data was taken from any of the control programs discussed in this report. As a supplement to the qualitative evaluation given below, a series of photographs were taken and are presented in Appendix D. The photos are time exposures of controlled-simulated traffic as it appears on the displays used in the experiments.

A. En route Control

One important factor in any control system employing feedback, is stability. From our study of the PICON series of control programs, it appears that several different types of instability can occur, each of which is difficult or impossible to relate to one comprehensive definition of stability. This is undoubtedly due to non-linearity in feedback, which is probably a permanent property of air traffic control. A comprehensive definition was assumed, but the study of stability was not confined to the definition as described below.

The control system is stable (by definition) if n flights converging on the same point will eventually reach a non-changing state of all holding; one of the flights must be holding at the point of convergence.

Using the above definition, stability was studied and found to vary significantly as a function of parameters, the most influential of which were θ_{VH} and T_{cmH} . (If a flight is turned to a heading that is more than θ_{VH} different from the straight-line heading to destination, it will be held. The future proximity parameter is T_{cm} ; any flight that is holding will not be released until future proximity is greater than T_{cmH} where T_{cmH} is greater than T_{cm} .) PICON is stable if θ_{VH} is less than 90° and T_{cmH} is twice T_{cm} .

Stability (as defined above) was not studied extensively because both of the influential factors are the results of approximations. It was observed that in normal operation, traffic behavior is less dilatory if parameters are chosen so that the system is stable (as defined above).

Another type of instability has been observed which apparently is independent of the comprehensive definition given above. En route control is characterized by a "chain reaction" in heading change which may propagate rapidly through several flights. Stability may be defined in terms of the propagating change as follows: the chain reaction is stable if any one of the heading changes is smaller than the preceding heading change. In PICON, the chain reaction is not always stable, in which case, the heading changes build as they propagate, until a turn of greater than 90° occurs. At this point the chain reaction is terminated since the 90° turn is the condition for en route hold. It has been observed that stability in the chain reaction is strongly linked to turn direction. If, for example, several flights are roughly in a line in overtake configuration, the chain reaction can be unstable if all turn in the same direction; on the contrary, it will be stable if one or more of the flights turn in the opposite direction from all of the others. In PICON, choice of turn direction is "that which requires the shortest turn" at the instant of calculation; this rule is essentially an expedient approximation.

Several other minor types of instability have been observed, and either corrected by changes in logic or traced to approximations. Except as described above, stability has not been studied carefully in its relation to normal operation.

In normal operation of the PICON series of control programs, traffic is observed to be maintained safe but not optimum. In almost all cases, optimization of traffic flow requires route planning and analysis, and, as indicated in the report, optimization was not considered because of insufficient memory space. The three most important defects of PICON en route control are discussed below; all three would be eliminated or improved by a set of optimization logic.

The most important shortcoming in PICON is failure to find a "best no-conflict heading." It will be recalled that the logic of collision avoidance was such that if θ_s (heading to station) is in conflict, the flight is maintained on θ_p (present heading, not in conflict) regardless of how θ_p compares with θ_s . This logic is inadequate in that θ_p will frequently be a heading that was chosen to avoid a conflict that occurred several minutes in the past. As such it is most likely an improper (but safe) heading in the current air situation. This event occurs frequently in normal operation; two examples are shown in the photographs of Figs. 28 and 29 of Appendix D. In Fig. 28 in particular, one of the flights (the northern-most) was maintained on take-off heading which obviously has no relationship to the conflict that exists along θ_s . A solution for this problem consists of designing another sorter intended to gather information from which a "best no-conflict heading" can be derived. Since such a heading depends on route planning for optimization, both optimization logic and the new sorter should contribute to the calculation of the best heading. Nothing was done on this problem because of system limitations.

Still a third type of sorter is necessary to solve another outstanding problem in PICON. This sorter would be called the "conflict resolution sorter" and would be designed to eliminate the pairing rule from conflict resolution. By the pairing rule, only the nearest conflict is avoided, as if no other

flights existed; even the turn direction is based on the nearest or primary conflict. It has been observed that traffic is frequently dilatory in heavy congestion, as the pairing rule switches the primary conflict from flight to flight. In addition, the limited choice of turn direction frequently causes a flight to turn toward congestion rather than away from it, in avoiding a primary conflict. Both of these undesirable effects could be eliminated by a resolution sorter operating in conjunction with optimization logic. The resolution sorter would gather information on all possible secondary conflicts (secondary conflicts are those generated in turning to avoid the primary conflict) and turn direction could be chosen to favor an optimum route.

The rules for entering and leaving en route holding patterns are also inadequate. A flight is instructed to enter a holding pattern if its present heading is greater than 90° different from the heading to its destination; a flight will leave a holding pattern if its heading to destination is clear of conflicts within the limits set by the proximity condition ($t_c \leq 20$ min). Though it does not occur often, it has been observed that the "90 $^{\circ}$ " point can occur such that a few moments more at θ_p (present heading) would result in a clear path to destination. By the rule, the flight must be held, and in so doing, suffers additional delay. Exit from hold is safe but can be extremely inefficient. If, for example, a flight is held such that there is a busy air terminal between the flight and its destination, it can be held indefinitely. Since hold is, indeed, a delay maneuver, control should choose the maneuver only in response to a delay analysis and exit from hold should be based on route planning which, of course, shall enable circumvention of congestion. Delay analysis and route planning are both part of optimization logic.

B. Approach and Terminal Control

In PICON, approach area traffic has priority over en route traffic; this is achieved by including approach traffic in the sort whenever the reference aircraft is an en route flight, but rejecting en route traffic from the sort whenever the reference flight is under approach control. This property causes traffic to cross the approach-area boundary well dispersed, which, in turn, results in good dispersal of aircraft within the approach area. If, for example, a fast aircraft in en route area is in overtake configuration with one in approach area, the fast aircraft will be turned, before entering approach area, so that overtake is impossible. If the fast aircraft is ultimately held in approach area, it will then be held much closer to an entry point than if it had continued on the original overtake path.

All conflicts in approach area are resolved by holding and all traffic not holding will be vectored directly to respective entry points. All holding patterns in approach area are oriented so that the inbound leg is on a radial emanating from the entry point. Minimum safe separation is specified by these factors, since each flight must be sufficiently isolated so that it can be safely held at any time. From observation of the system in operation, the rules appear to be adequate except as noted below. Traffic behaves particularly well under conditions of heavy congestion.

Unfortunately, spacing that will enable safe holding of any approach-area flight at any time, is too wide to permit a "no-gap" landing sequence. For this reason, four entry points were chosen and from observation, this number is more than adequate. When the traffic load is heavy, all sectors and all entry points are always busy, and traffic flow is quite smooth. There is even a tendency for traffic to become evenly distributed within the four sectors of approach area, since en route control steers traffic to avoid the congested

sectors (by the priority rule referred to above).

Oddly enough, there is a minor problem in the PICON approach logic when traffic is light, but only if all traffic is arriving from the same direction. In this case, all flights can enter the same sector, and subsequently arrive at the entry point with spacings determined by minimum-safe-holding separations. This spacing guarantees gaps in the landing sequence; an example of this is shown in the photograph of Fig. 26. If the four sectors of approach area are maintained intact as described in the text, there is no solution for this problem since safety must be maintained.

Another minor problem of approach control in PICON is the following: It is possible, by the approach logic used here, for a holding flight to remain holding indefinitely depending on other approach-area traffic. This can happen if a slow aircraft is held along an edge of a sector about 15 miles from the entry point. The holding flight can have the effect of creating an open high-speed lane in the other half of the sector; the lane will remain open as long as high-speed traffic can pour through it. The problem could be solved by setting a time limit on approach-area hold or by modifying the logic. Neither solution was attempted because of system limitations. In operation, the situation exists infrequently.

Terminal control in PICON leaves little to be desired within the experimental framework. Spacings are well maintained and traffic flow is smooth. As described in the text, minimum spacing in the terminal area is determined by the speed range of terminal traffic. In PICON, this spacing is approximately 1 mile and will subsequently occur any time that a "highest speed" flight and a "slowest speed" flight are holding at the outer pair of entry points. (In PICON, highest and lowest speeds are 206 mph and 112 mph respectively.) The

safe speed-range of PICON, is inadequate for a real system but there are many possibilities for expansion of this range. This was not studied further because of system limitations.

C. Computation Times

PILOT was programmed to simulate up to 64 airplanes. This figure was chosen early in the work on the basis of control computation-time estimates. In practice, it has been found that PICON begins to lose control with 22 to 25 aircraft in the system. Control loss occurs when ILLIAC needs more than 3 seconds to process all traffic. (In actual operation, the clock signal instructs PICON to "stop what you are doing and start over." This was necessary to insure terminal sequencing every three seconds.) Since these figures appear to be unfavorable for a fully automatic real system, some comments regarding the very low saturation level in PICON are in order. The three most time consuming factors in PICON are drum transferring, simulation of traffic, and the iterative process used in conflict resolution. The iterative process has been described in detail in the text and so will not be repeated here.

PICON is a 1500 word program, half of which is stored on the ILLIAC drum. To execute the entire program once requires two drum transfers which consume .5 seconds each. Since the program must be executed once every clock interval, 1 of the 3 seconds available is used for drum transferring.

Simulation of aircraft is the next most time-consuming part of PICON. PILOT consumes two full frames, one for decision making and one for executing all turns. It takes TASC between 2.5 and 10 milliseconds to transfer one target to ILLIAC. The time spread is determined by how much work ILLIAC must do on each target. It follows that 50 targets in the Cornfield System will consume .25 seconds on the average just to get into ILLIAC, not to mention

the time to process targets in PILOT simulation logic.

In PICON, it was necessary to program all of the "tools" described and several others of lesser importance. The tools then appear in PICON as 10 or 20 word sub-routines. Execution of subroutines is, in general, much more time-consuming than it is to execute the same functions with electronic devices. Some typical execution time relationships are listed below:

<u>Function</u>	<u>PICON</u>	<u>Automatic System Typical Time</u>
ΔJ_i check	690 microseconds	1 microseconds
ΔJ_i store	890 "	2.5 "
Class switching	416 "	2 "
Plant a heading	700 "	2.5 "
Plant a class	845 "	2.5 "
Access to 2 targets	3140 "	5.0 "
Sort one against 50	300,000 "	575 "

One additional factor pertaining to computation time is worthy of note. All general purpose digital computers can only do one things at a time (with some minor exceptions). If a general purpose machine were used to do the job described in Part I of this report, it would have to devote roughly 1/4 of its time to each of the four basic functions (control, display, track, and communication). This is undesirable of course. It was partly for this reason that the configuration shown in Fig. 1 has been chosen as the best for an automatic system. Each of the four logic sections depicted in Fig. 1 is a separate digital device roughly equivalent to a small digital computer. In this arrangement, the four basic functions are not executed in time sharing but instead are executed simultaneously in parallel. The same technique of executing functions in parallel can be carried into control logic to further reduce computation time.

D. Further Developments

In view of the fact that the first mid-air collision between two IFR (controlled) flights has already occurred, it appears to be imperative that the existing manual control system be improved. It is generally accepted that major improvements lie in automation, and it is our contention that the most effective improvements consist of automating decision making. This being the case, the work begun here should be continued. Needless to say perhaps, one of the goals in continuation should be to find better tools and more effective control logic. This, of course, includes devising a system of optimization logic.

Switching from an experimental to a real system necessarily includes elimination of all assumptions in the experimental system. To the best of our knowledge, the most difficult assumption to eliminate is "perfect automatic tracking." Although this is primarily an engineering problem, some comments regarding elimination of this assumption are made below.

All of the work done and reported on automatic tracking has been on the design of computers to be used in the military environment. In the military environment, the target of primary interest (hostile) is most uncooperative, and this factor reflects strongly in the design of a tracking computer. On the contrary, air traffic in the civilian environment is cooperative, and, in similar fashion, this knowledge can be exploited in the design of a tracking computer. To mention a few examples: controlled traffic will not turn unless instructed, will maintain altitude, and will continue on course through long periods of radar fade; all controlled traffic will be well spaced with respect to all other traffic and if necessary, crossing tracks can be totally avoided by control. In addition, controlled traffic (and possibly all traffic) of the

future will provide radar return enhancement along with auxiliary information in binary form (Project Beacon Report, Oct. 1961). In any case, failure to exactly achieve the goal of perfect tracking can be compensated by human intervention.

Probably the second most difficult problem is the elimination of the assumption "winds aloft are zero." Upper winds of any velocities and distributions are easy to simulate, but limited system capacity prohibited simulation in PICON. To compensate for wind, the control computer will have to know the wind, and the most convenient mechanism for gathering such information is air traffic itself. The problem is complicated by the fact that heading-following errors in traffic are random. It is important to note that erroneous knowledge of wind does not jeopardize the safety of controlled traffic (because of ΔJ_i logic); it only makes the problem of optimization more difficult.

The logic of terminal control has one important shortcoming, namely, safety is not assured if $K > 1.84$ (The ratio of maximum speed to minimum speed is defined as K . In PICON, $s_{\max} = 206$ mph and $s_{\min} = 112$ mph.) Fortunately, several possibilities are handy for solving this problem, though none were tried in PICON. There is considerable flexibility in the number and location of entry points, for example; in addition, there is no reason why the condition " $K < 1.84$ " can't be one of those necessary for selecting a flight for sequencing. The latter solution has the advantage of flexibility in minimum spacing, i.e. K can be chosen as 1.5 if desired in which case, minimum spacing would be increased to 4 or 5 miles.

Elimination of the "one altitude" assumption will have a mitigating effect on many of the problems encountered in this work. The assumption was chosen partly because of system limitations and partly because it was felt that the choice would perpetrate the most difficult problems in automatic

control. In reality, there are two general problems in use of altitude:

1) initial choice of cruising altitude and 2) change of altitude particularly in arrival and departure. There are several trivial solutions to the first problem and several interesting and not so trivial possibilities for solution of the second problem.

Though not specifically listed as an assumption in this work, it is obvious that there is one assumption concerning human controllers, i.e. humans have no ability to influence or participate in decision making. While this is certainly a practical assumption at this level of work, it may not stand in the best real system. Addition of human decision makers can be justified in two cases: 1) if it results in a significant reduction in equipment with no loss in quality of control or 2) if it results in better control with insignificant increase in equipment. The most difficult problem obstructing the use of humans in any ordinary way is the cause and effect chain reaction inherent in air traffic control. The chain reaction is particularly prevalent in en route area. In a terminal area serving a single runway, the chain reaction can be limited to four or five aircraft with trivial effect on traffic in approach and en route areas. The problem of how to use humans in a man-machine control system is difficult and remains to be considered.

V. APPENDIX A. DERIVATION OF VARIABLES

The general expression for position of an aircraft is given below in rectangular coordinates:

$$x_i = x_{0i} + u_i t,$$

$$y_i = y_{0i} + v_i t,$$

where (x_i, y_i) is the position of the i th aircraft as a function of time, (x_{0i}, y_{0i}) is present or starting position and t is time.

Assuming a reference aircraft (subscript 1), the general expression for the distance between the reference aircraft and the i th aircraft is

$$d(t) = \sqrt{[(x_{01} - x_{0i}) + (u_1 - u_i)t]^2 + [(y_{01} - y_{0i}) + (v_1 - v_i)t]^2}$$

from which $t_{ci}(t)$ can be written,

$$t_{ci}(t) = \frac{\sqrt{[(x_{01} - x_{0i}) + (u_1 - u_i)t]^2 + [(y_{01} - y_{0i}) + (v_1 - v_i)t]^2}}{s_1 + s_i}$$

Minimizing the above expression yields t_{mi} and substituting t_{mi} back into the above expression yields t_{cmi} . While t_{mi} is the time interval necessary for t_{ci} to shrink to t_{cmi} , it is also the time interval necessary to reach closest passage since

$$t_{ci}(t) = \frac{d_i(t)}{s_1 + s_i}$$

in which $(s_1 + s_i)$ is constant.

VI. APPENDIX B. CONFLICT RESOLUTION SPECIAL CASE

The variables used below can all be identified from Fig. 19. In addition,

$$u_i = s_i \sin \theta_i,$$

$$v_i = s_i \cos \theta_i,$$

$$y_{02} - y_{01} = d_{12} \cos \theta_0,$$

$$x_{02} - x_{01} = d_{12} \sin \theta_0,$$

$$b = \frac{s_2}{s_1} \quad \text{by set,}$$

$$t_c = \frac{d_{12}}{s_1 + s_2},$$

$$t_m = \frac{(x_{02} - x_{01})(u_1 - u_2) + (y_{02} - y_{01})(v_1 - v_2)}{(u_1 - u_2)^2 + (v_1 - v_2)^2}.$$

It is desired to calculate θ_1 as a function of θ_2 such that

$$t_m \leq t_c.$$

By direct substitution,

$$\frac{(x_{02} - x_{01})(u_1 - u_2) + (y_{02} - y_{01})(v_1 - v_2)}{(u_1 - u_2)^2 + (v_1 - v_2)^2} \leq \frac{d_{12}}{s_1 + s_2}.$$

Substituting again from the list of trigonometric equations above and at the same time setting $\theta_0 = 0$,

$$\frac{s_1 \cos \theta_1 - s_2 \cos \theta_2}{(s_1 \sin \theta_1 - s_2 \sin \theta_2)^2 + (s_1 \cos \theta_1 - s_2 \cos \theta_2)^2} \leq \frac{a}{s_1 + s_2}$$

Setting θ_0 to zero has the effect of aligning the two flights on a north-south line but does not change the properties of θ_1 as a function of θ_2 . This is more clearly seen from Fig. 19. Expansion of the squares and simplification of the above expression yields

$$A \cos \theta_1 + B \sin \theta_1 \leq C,$$

in which

$$A = 2ab \cos \theta_2 + (1+b),$$

$$B = 2ab \sin \theta_2,$$

$$C = (1+b)(a+b \cos \theta_2).$$

Solving for θ_1 ,

$$\cos(\theta_1 - \alpha) \leq \frac{C}{A + \sqrt{A^2 + B^2}}$$

where $\alpha = \arctan \frac{B}{A}$.

Observation of this expression leads to the conclusion that a solution for $\cos(\theta_1 - \alpha)$ does not always exist. Therefore an existence equation was derived as follows: Since

$$-1 \leq \cos(\theta_1 - \alpha) \leq +1,$$

A solution exists only if

$$-1 \leq \frac{C}{A + \sqrt{A^2 + B^2}} \leq +1$$

or if
$$\frac{|C|}{\sqrt{A^2 + B^2}} \leq 1.$$

Expanding yields the following existence equation:

$$(1 + b) |a + b \cos \theta_2| \leq \sqrt{4a^2b^2 + (1+b)^2} + (1+b) 4ab \theta_2.$$

Setting "a" to 1/2 (because "a" is 1/2 in PICON) and using ordinary methods, it is found that a solution exists for $\theta_1 = f(\theta_2)$ only if $b \leq 1$. If $b > 1$, a solution exists only for certain θ_2 .

The above mentioned effect occurs because t_{mi} is mostly determined by the faster of two closing aircraft. Failure in existence means that there is no θ_1 such that $t_m \leq a t_c$. The calculated effect has been verified by experiment.

In PICON, this problem was left as it stands mostly because there was not time to experiment and to devise a better solution. In operation, quantization error causes trouble relatively infrequently and the probability of finding "no solution headings and speed ratios" is even lower. Successful operation (to the extent studied) was achieved in spite of this condition.

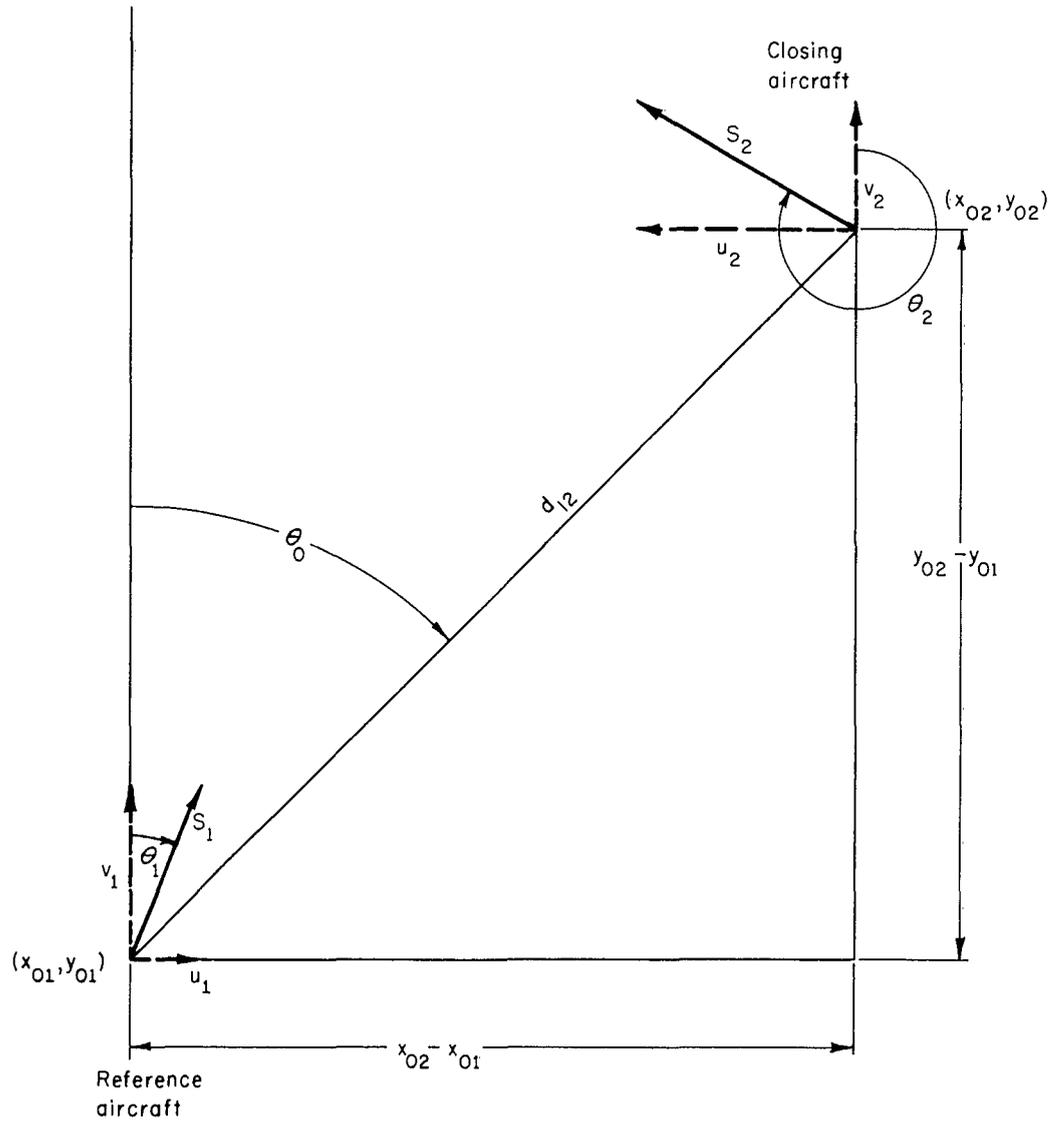


Fig. 19. Definition of Variables.

VII. APPENDIX C. ESTIMATION OF LANDING TIME

The landing time is a function of the average speed and the length of the path to be traversed; its estimate, therefore, is derived from the estimates of speed and path length.

A. Estimation of Average Speed

An estimate of the average speed for planes on the final approach path can be obtained from the formula

$$(1) \quad s_{av} = \frac{K_1 s_{exp} + K_2 s_{obs}}{K_1 + K_2},$$

where s_{exp} is terminal area speed prescribed for a given type of aircraft and s_{obs} is ground speed derived from acquired data, and K_1 and K_2 are parameters. In all simulation runs, $K_1 = K_2 = 1$ setting was used.

For lack of space in computer memory, the prescribed speed s_{exp} was assumed to be constant throughout the terminal area. The extension to cases in which it varies over the path appears to be relatively simple: The path can be divided into any desirable number of segments and s_{exp} taken to have a different value for each segment.

B. Estimation of Path Length

The path length was estimated for the tangential approach paths whose general form is shown in Fig. 12. The line of the current heading of an aircraft was extended to intersect ILS at P_0 at an angle α (Fig. 20). It was assumed that the pilot would maintain exactly the heading he was instructed to fly and, consequently, the last heading given to the pilot was taken as the current heading.

The rate of turn was taken to be $180^\circ/\text{min}$ for all aircraft. The radius of curvature of the turn-on arc becomes thus a function of the speed of the aircraft only and is given by

$$(2) \quad r = \frac{S}{60\pi} \text{ mi.}$$

If the turn-on arc is tangent to ILS and the current heading extension at P_2 and P_1 respectively (Fig. 20), the length of the segment $\overline{P_0P_1}$ is given by

$$(3) \quad \overline{P_0P_1} = \frac{r}{\tan \frac{1}{2}\alpha}$$

and the length of the arc P_1P_2 by

$$(4) \quad P_1P_2 = (\pi - \alpha)r,$$

where α is expressed in radians.

The total distance along this path from P to P_3 , where P is the present aircraft position and P_3 the turn-off point on the runway, is obtained from

$$(5) \quad d = \overline{PP_0} + \overline{P_0P_3} - 2 \overline{P_0P_1} + P_1P_2.$$

To shorten computing time, the straight line distance R between any two points (x_1, y_1) and (x_2, y_2) was approximated by

$$(6) \quad R \approx |x_1 - x_2| + |y_1 - y_2| - \frac{|x_1 - x_2| |y_1 - y_2|}{\frac{7}{8}(|x_1 - x_2| + |y_1 - y_2|)}$$

Formula (6) gives the required distance with less than 1 o/o error.¹

Note that using (5) to obtain the total length of the path requires the calculation of the coordinates of only one intersection point, namely P_0 .

C. Estimation of Landing Time

The estimate of landing time t_1 of a plane at P, with speeds s_{exp} and s_{obs} , proceeding on a heading which forms an angle α with ILS, and pursuing a tangential approach path, is clearly

$$t_1 = \frac{d}{s},$$

where d is given by (5) and s by (1).

1 cf. James Snyder and L. Fosdick, "On the use of a high speed general purpose digital computer as the control element in a surveillance and control system," CSL Report R-74, pp. 253-5.

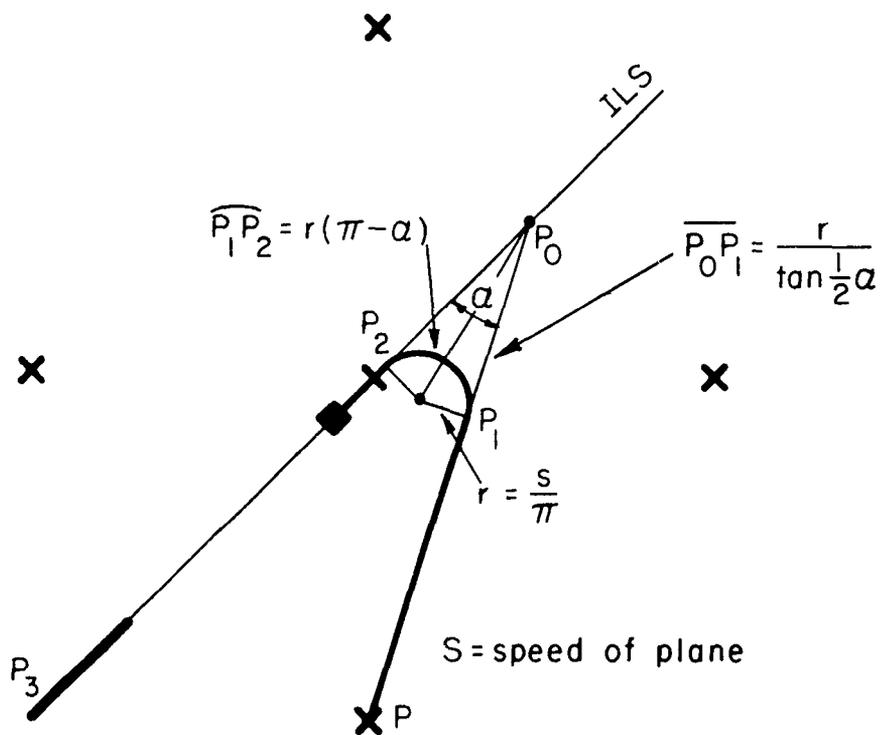


Fig. 20. Estimation of a Tangential Approach Path.

VIII. APPENDIX D. PHOTOGRAPHS

The following photographs are time exposures of a display console, as simulated air traffic operates in real time. The dimension of time is added in the explanatory material under each photo. In those photographs in which flights are identified by alpha-numeric characters, the middle digit of identification is also the aircraft type digit. The following table of type vs middle digit is presented as a supplement to the photographs.

<u>Middle Digit</u>	<u>Type of Aircraft</u>	<u>Speed Range</u>
0	tri-pacer	112-131 mph
1	Cessna 310	131-206 mph
2	electra	168-355 mph
3	jet	206-542 mph

There are three sets of photos illustrating 1) automatic approach and terminal control, 2) automatic en route control, and 3) manual approach and terminal control. These three sets of photos are identified by the program being used, i.e., PICON, PICON-ER, and PILOT.

A. PICON

The following six photographs (Figs. 21 through 26) are time exposures of one of the displays while PICON was controlling ten simultaneous arrivals. In this particular run, the only en route control activity not shown in the photographs was that effecting jet 138 arriving from the southeast. The jet started from a point due southeast of the outer marker but was turned 30° toward the west to avoid overtaking 115 (Cessna 310). After the overtake was resolved, the jet was turned toward the outer marker to arrive as shown in Fig. 21. The first photograph (Fig. 21) was a 12 minute exposure and all

following photos were 4 minutes each. Time is taken to be zero at the start of the run. The active runway is 22 (220°) and the runway is displayed as 5 spots in all cases (the runway shows clearly in Fig. 21). Entry points and outer marker are also displayed.

The following table of arrivals was tabulated from the run shown in the photos, Figs. 21 through 26.

<u>Type and Identity</u>	<u>Arrival Times</u> <u>min:sec.</u>	<u>Time Between</u> <u>Arrivals</u> <u>min:sec.</u>
tri-pacer 109	15:50	----
electra 12L	17:50	2:00
Cessna 310 (115)	20:40	2:50
Cessna 310 (117)	23:15	2:35
jet 130	24:51	1:36
tri-pacer 10J	27:56	3:05
jet 138	29:38	1:42
electra 124	31:44	2:06
Cessna 310 (112)	34:26	2:42
electra 12K	37:11	2:45

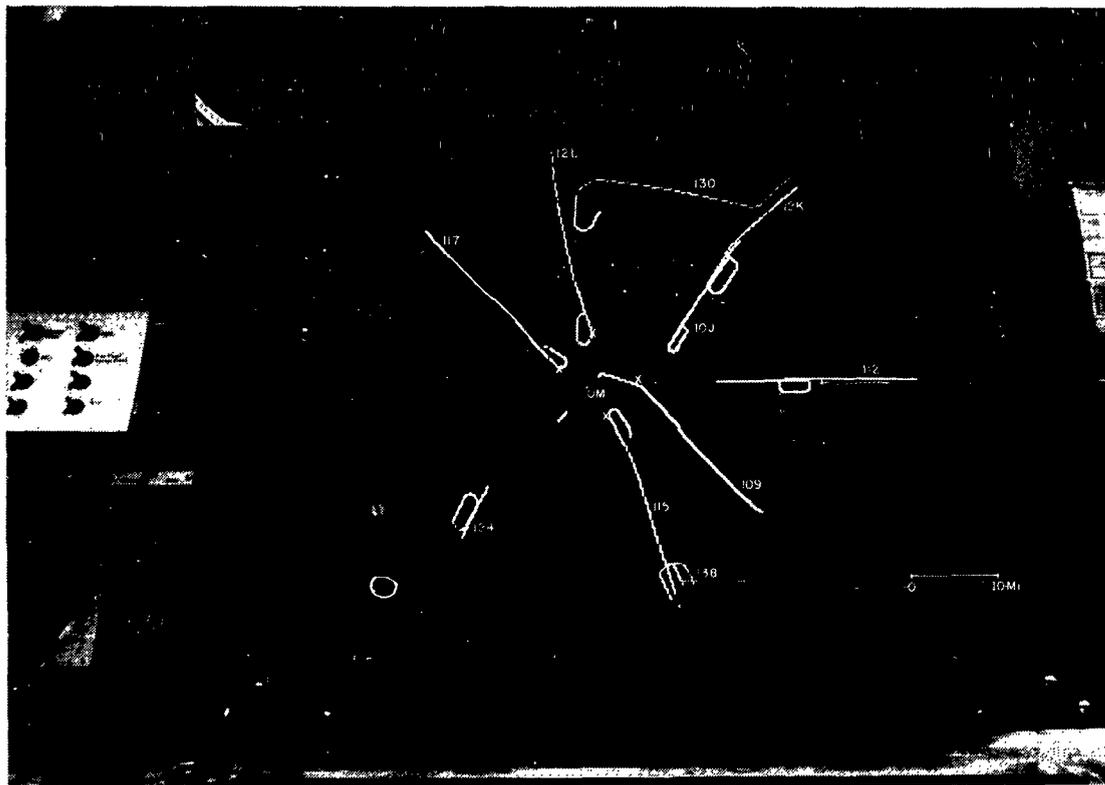


Fig. 21. This is a 12 minute exposure recording all events in the time interval 00 min. to 12 min. No traffic has landed as yet but tri-pacer 109, the first flight to reach an entry point, has just turned onto the ILS. The two tri-pacers (109 and 10J) were scheduled to reach the entry point simultaneously. PICON chose to hold 10J, which caused electra 12K to be held, which in turn caused jet 130 in en route area to be turned to the west. The jet proceeded WNW until it was turned into approach area behind electra 12L. The jet was held immediately because of 12L holding at the entry point. Actually, at time 12:00, 12L had just been selected for terminal sequencing behind 109, which freed the entry point for 130. In Fig. 21, 130 is just beginning to turn toward the entry point. The holding flight in the southeast is also a jet (138). The double track in the pattern is due to speed reduction executed during the first turn. In the southwest, electra 124 has been held because of 117 holding at the west entry point. In Fig. 21, 124 has executed the hold pattern once and is being moved closer to the entry point. In the east, 112 (Cessna 310) has been held in response to 109 and 10J both of which were closer to the entry point at the time the decision was made. The Cessna executed the hold pattern once and then was vectored toward the west entry point as 109 proceeded beyond the entry point.

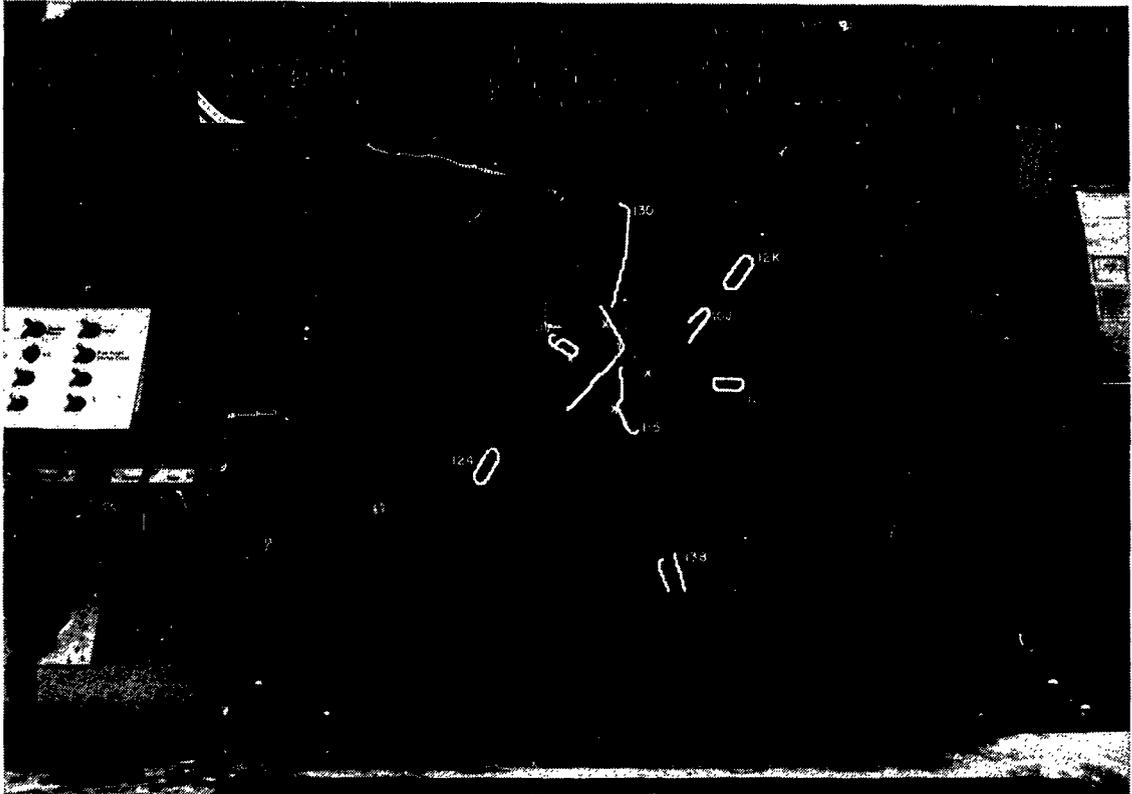


Fig. 22. This is a four minute exposure recording all of the events of the time interval 12 min. to 16 min. At the end of this exposure, tri-pacer 109 had landed (at 15 min. 50 sec.) and electra 12L is on the ILS over the outer marker. Jet 130 has been steered toward the north entry point vacated by 12L but had not reached it by time 16 min. In the northeast and east, tri-pacer 10J has been vectored toward the entry point while electra 12K and the Cessna 310 (112) continue to hold. In the southeast, 115 has been selected for terminal sequencing and has been vectored toward the ILS to follow 12L in the landing sequence. In response to this, jet 138 has been taken off hold and is being vectored to the entry point vacated by 115. In the west and southwest, the Cessna 310 (117) has been selected to follow 115 and though not apparent from the photo, electra 124 has been vectored toward the entry point being vacated by 117.

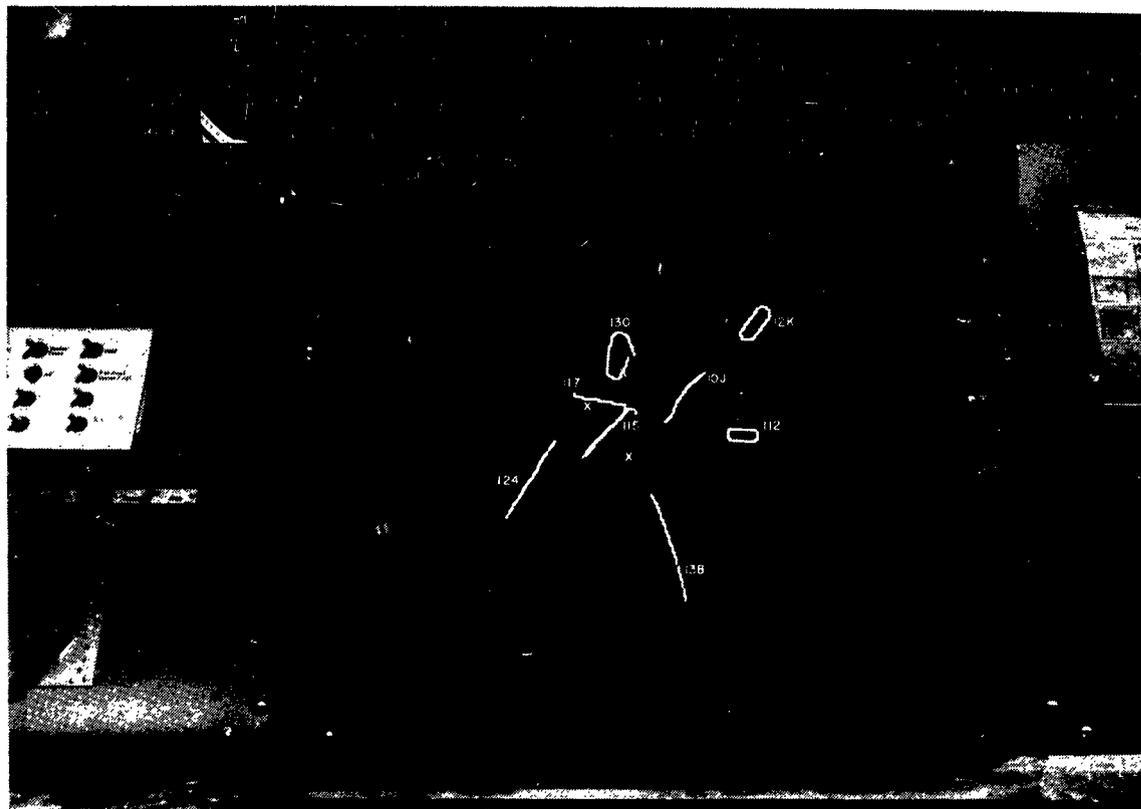


Fig. 23. This is a four minute exposure recording the events of the time interval 16 min. to 20 min. Electra 12L has landed at 17 min. 50 sec., and the two Cessna 310's are in final approach, 115 just touching down and 117 on the ILS approaching the outer marker. In the east and northeast, the electra and the Cessna 310 continue to hold while tri-pacer 10J has just reached the entry point. In the north, jet 130 has been selected for terminal sequencing behind 117. Jet 138 and electra 124 in the south and southwest proceed toward their respective entry points.

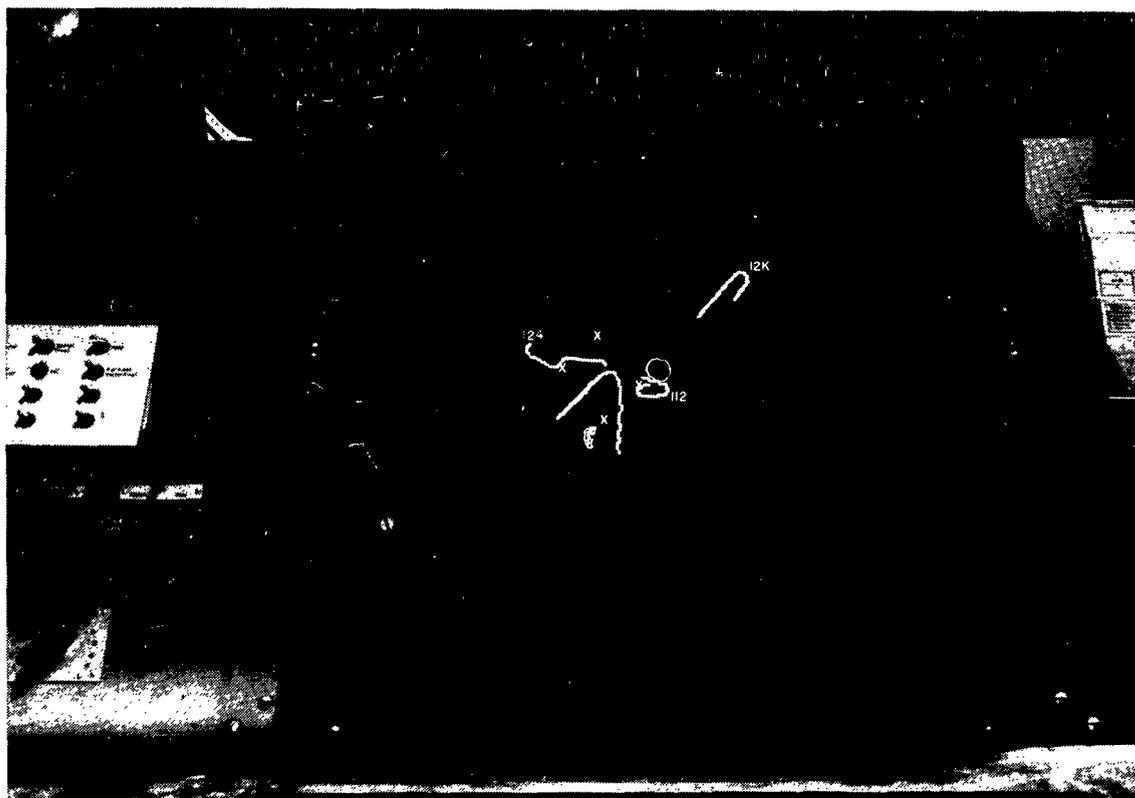


Fig. 25. This exposure covers the time interval from 24 min. to 28 min. Jet 130 landed early in the exposure at 24 min. 51 sec. Tri-pacer 10J followed at 27 min. 56 sec. Jet 138 has been vectored into the ILS from the south entry point to be landed behind tri-pacer 10J. At the end of the exposure, 138 was almost over the outer marker. In the meantime, electra 124 has been vectored into the ILS from the west entry point for landing behind 138. In the east and northeast, the Cessna 310 (112) has been selected to follow 124 in the landing sequence and electra 12K has been taken off hold and vectored toward the entry point vacated by 112.

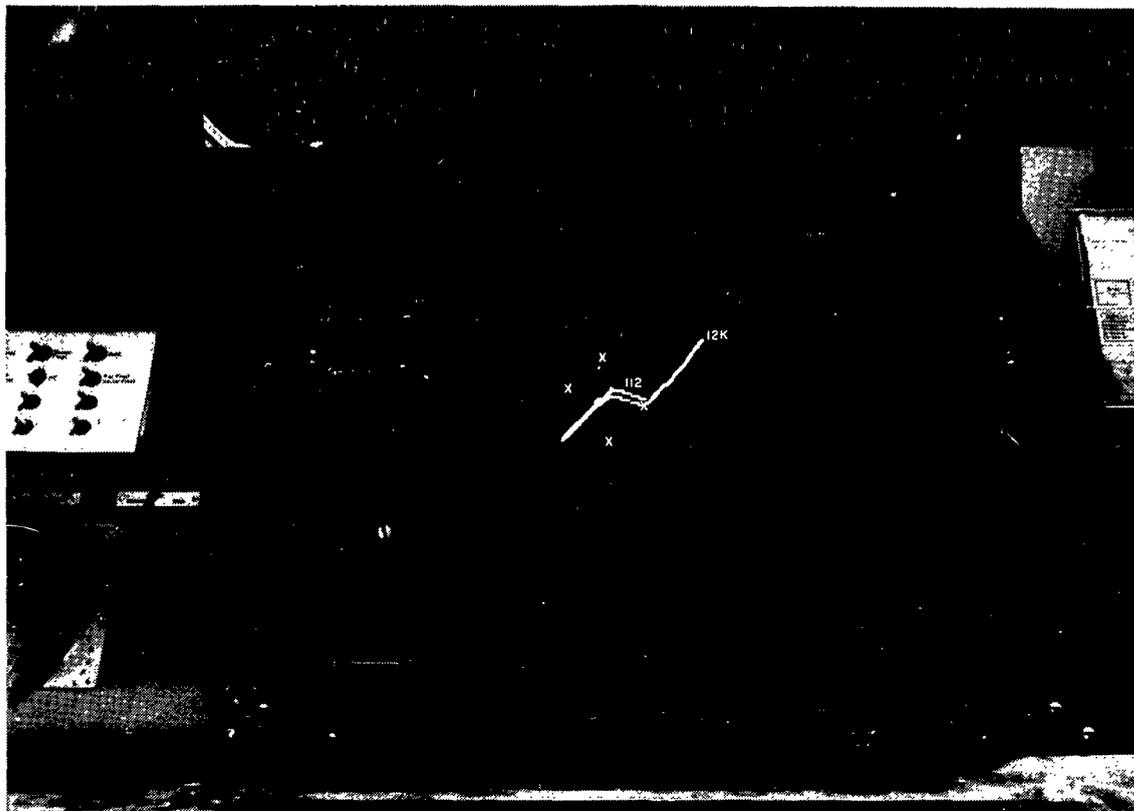


Fig. 26. This exposure covers all events in the time interval from 28 minutes to 34 min. 26 sec. at which time the Cessna 310 (112) was "clear of the runway." Jet 138 landed early in the photo at 29 min. 38 sec. followed by electra 124 at 31 min. 44 sec. followed in turn by the Cessna 310 (112) at 34 min. 26 sec. At the time of landing of 112, electra 12K was somewhat short of the outer marker producing the only unintended gap in the landing sequence throughout the run. This occurred because the approach area minimum spacing within any one sector must be so large as to prevent making up the gap in terminal sequencing. This point is discussed in detail in the text. The electra landed at 37 min. 11 sec. which is about 30 sec. later than necessary by the terminal sequencing rule.

B. PICON-ER

As described in the text, PICON-ER is intended to favor studying the en route control philosophy. The size of the control area has been increased to 512 miles on each side and terminal sequencing has been removed. The area simulated is the Chicago, St. Louis, Indianapolis area with ten major airports used in the simulation. These are listed below along with their respective identifiers to help orient the reader with the photographs.

<u>Air Terminal</u>	<u>Identifier</u>
St. Louis	STL
Quincy	UIN
Springfield (Ill.)	SPI
Decatur	DEC
Champaign	CMI
Moline	MLI
Rockford	RFD
Chicago	MDW
Indianapolis	IND
* Terra Haute	HUF

* A bit misplaced.

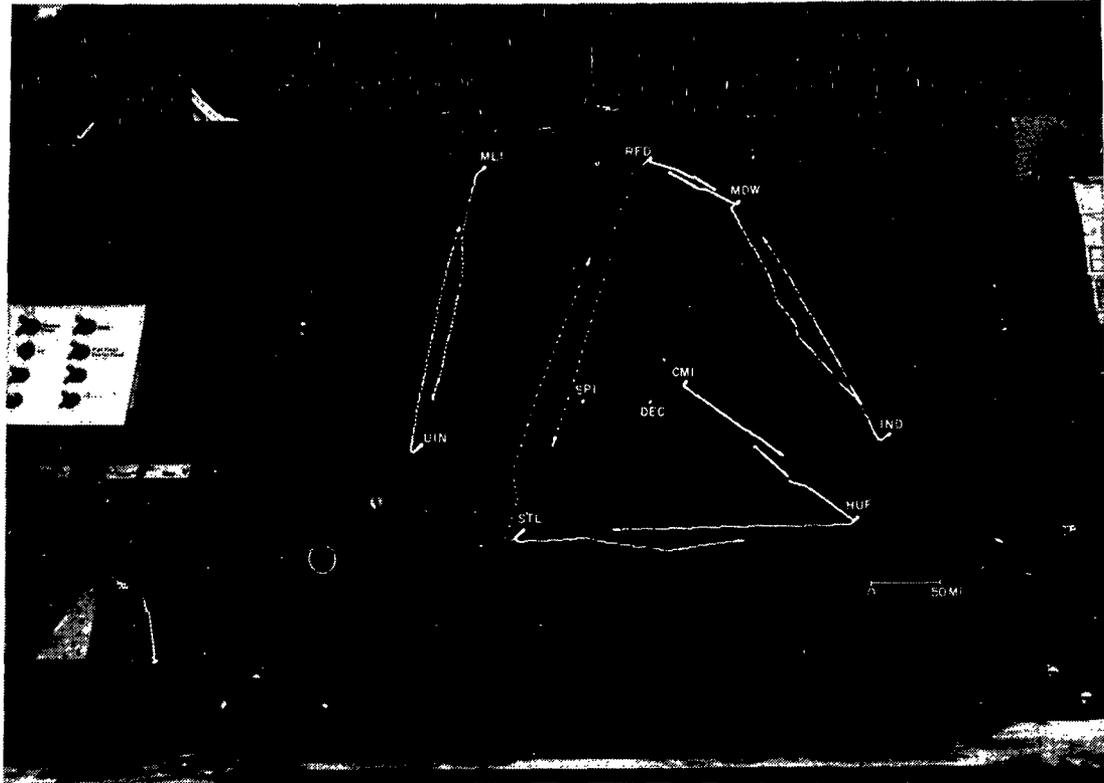


Fig. 27. This photograph is a 40 minute exposure in which all flights departed from their respective airports within the first three minutes of the exposure. All take-offs were toward the southwest and all destinations were intentionally chosen so as to create 6 non-interfering head-on conflicts. After 40 minutes, none of the flights had reached their destinations but all conflicts had been resolved and passed by. The photograph shows clearly the control action taken and it also shows the relative spacings maintained between the various types of aircraft involved. The table below shows which types of aircraft are on which routes.

<u>Route</u>	<u>Types of Aircraft</u>
Quincy - Moline (UIN-MLI)	electras
Rockford - St. Louis (RFD-STL)	jets
St. Louis - Terra Haute (STL-HUF)	electras
Champaign - Terra Haute (CMI-HUF)	Cessna 310's
Chicago - Indianapolis (MDW-IND)	electras
Rockford - Chicago (RFD-MDW)	tri-pacers

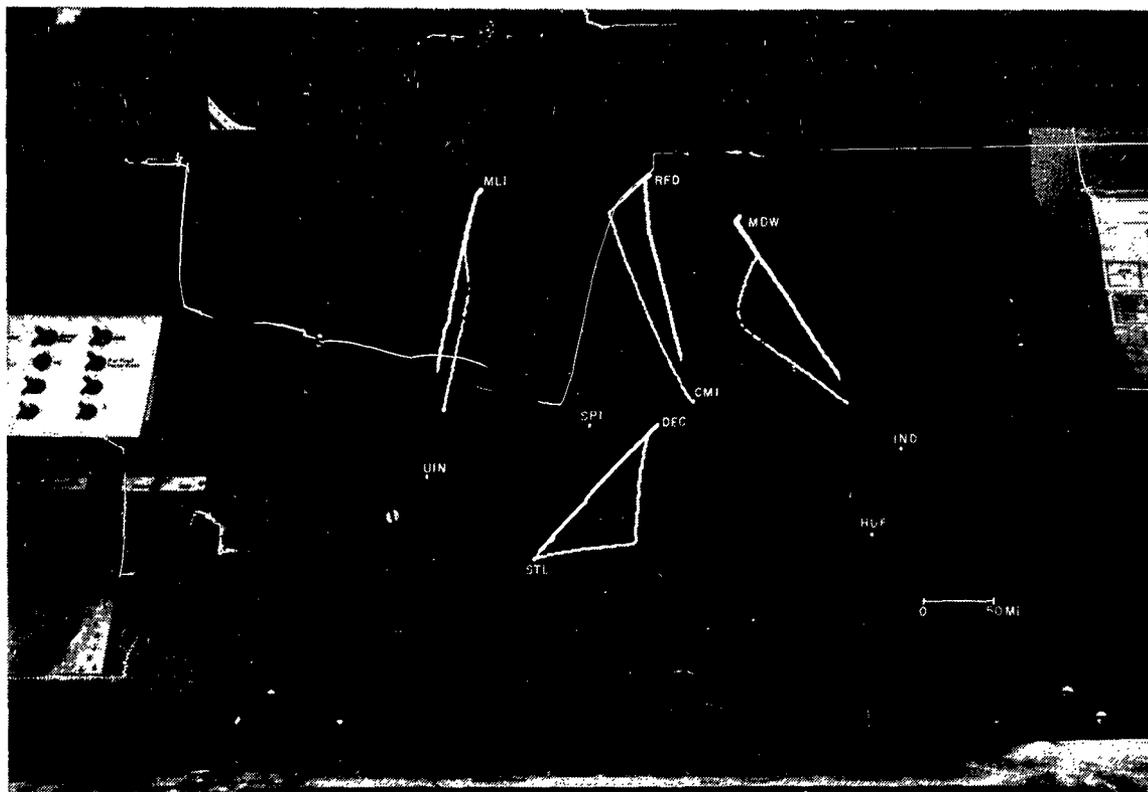


Fig. 28. This photograph is of approximately one hour exposure duration and shows four non-interfering overtakes. In all cases, the overtaken aircraft is a tri-pacer and the overtaker is either a jet or an electra. The photo is particularly instructive as there are two distinctly different effects presented. All overtakes are safe but only one (the southbound overtake from MLI to UIN) is optimum.

The devious routes taken in the non-optimum overtakes are due to the fact that only θ_p and θ_s (headings - present and to station) are used in the collision sort. As described in the text, if θ_s is in conflict and θ_p is not, the flight is maintained at θ_p . If the sorter calculated a "best no^p conflict" heading, the devious routes would not have occurred.

The optimum overtake is actually optimum by accident. At the time the conflict was discovered by ILLIAC, the tri-pacer was exactly half-way to its destination so that both the jet and the tri-pacer acted in the resolution. (The tri-pacer took off first and therefore its ΔJ_1 was set to $t_d/2$ - see ΔJ_1 Logic). Consequently, the tri-pacer was turned slightly toward the west with the result that the jet found θ_s clear and took a more optimum route.

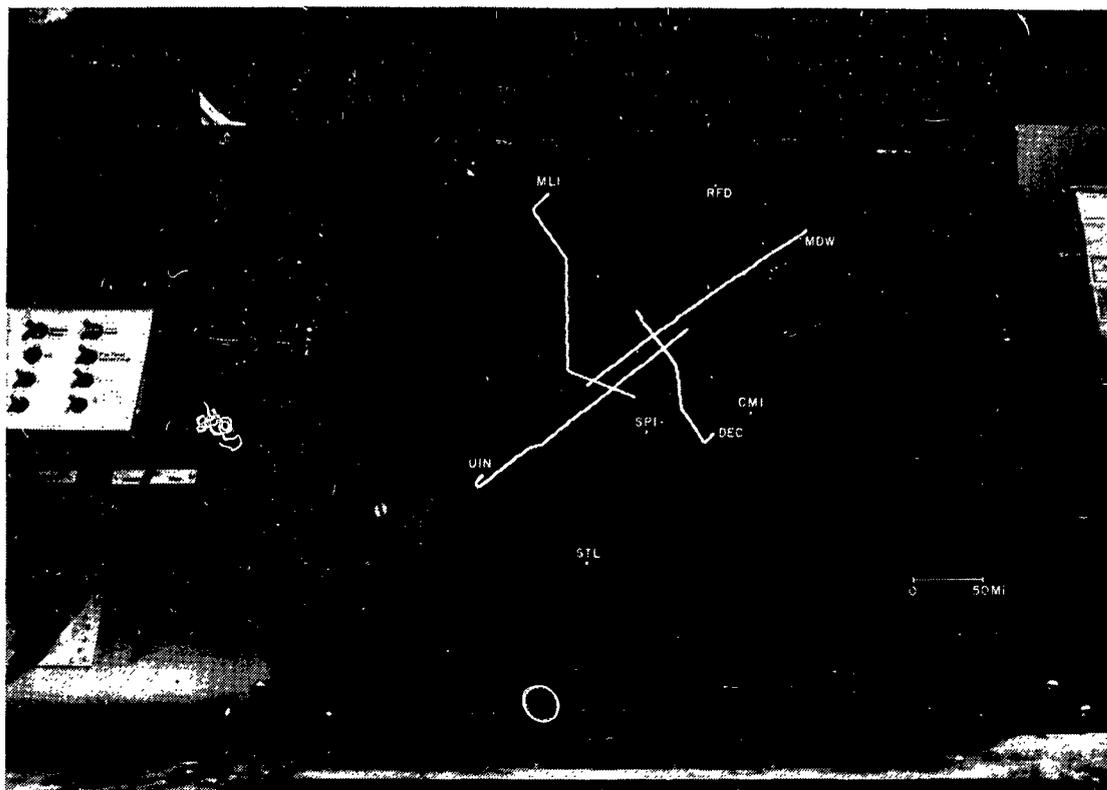


Fig. 29. The run duration and exposure is approximately 45 minutes. The flights originating at UIN, MLI, and MDW are electras and the flight originating at DEC is a Cessna 310. If the flights had proceeded unhindered, they would have passed very close to each other just NNW of SPI. At the end of the exposure, the multiple conflict has been resolved and passed, by all flights. Clearly the passage is safe but not optimum. The long southbound leg of the flight from MLI to DEC is due to the fact that the sorter does not calculate a "best no conflict" heading. Along the southbound leg, θ_s was always the conflict but θ_p was not; θ_p was a proper choice at the time the turn was made but thereafter is not necessarily a good choice.

C. PILOT

The next series of seven photographs (Figs. 30 through 36) are presented primarily as a matter of interest. The run does not constitute a controlled experiment nor are the comparisons with PICON rigorous or conclusive. As mentioned earlier, it is not our purpose to show that the automatic system can do a better job of sequencing than can human controllers.

The controller used for the run was an experienced pilot but totally without radar or controller experience. He was seated before a display identical to the one photographed. In addition to positional information, he was given identities and velocity vectors in the form of alpha-numeric characters and velocity noses. His control instructions were spoken as if into a microphone; in reality, instructions were relayed to ILLIAC via a keyset operated by someone else. All spoken instructions were of the form "130 vector 220^o," "12K vector final 270^o," or "115 hold."

The script used was the same as that used for the PICON photos (ten simultaneous arrivals) and jet 138 was given the same "out of view" vector as it received from PICON. Landing times were recorded and are tabulated below.

<u>Type and Identity</u>	<u>Arrival Time</u> <u>min : sec</u>	<u>Time Between</u> <u>Arrivals</u> <u>min : sec</u>
tri-pacer 10J	15:34	----
Cessna 310 (112)	16:39*	2:05
tri-pacer 109	20:43	4:04
Cessna 310 (115)	23:07	2:24
jet 138	26:13	3:06
electra 12L	29:04	2:51
Cessna 310 (117)	33:06	4:02
electra 124	35:36	2:30
jet 130	38:31	2:55
electra 12K	42:29	3:58

* Sequencing violation: 112 passed the outer marker 20 seconds before 10J was off the runway.

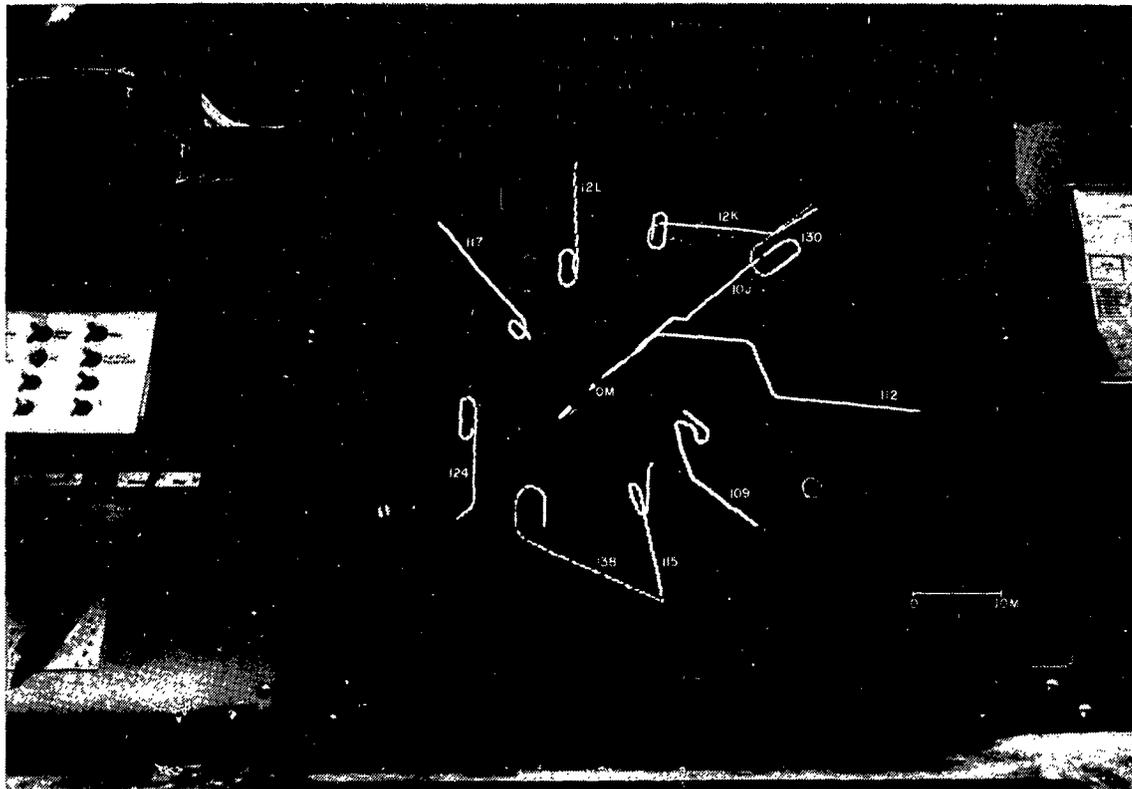


Fig. 30. This exposure covers all events in the time interval 00 minutes to 12 minutes. Early in the exposure, the controller had only three flights (10J, 109, and 112) on his display so he proceeded to sequence these three flights in the order 10J, 112, 109. Tri-pacer 10J was turned to the west, intercepted the ILS and proceeded to land. Cessna 112 was turned toward the northwest to increase spacing behind 10J, and 109 was held for approximately 2 min. before being turned to follow 112. In the south, Cessna 310 (115) was held for 3 minutes, then vectored north to follow 109 in the landing sequence. All other traffic was held except for electra 12K which was first vectored west. At the end of the exposure, two jets are holding, jet 138 in the south and jet 130 in the northeast.

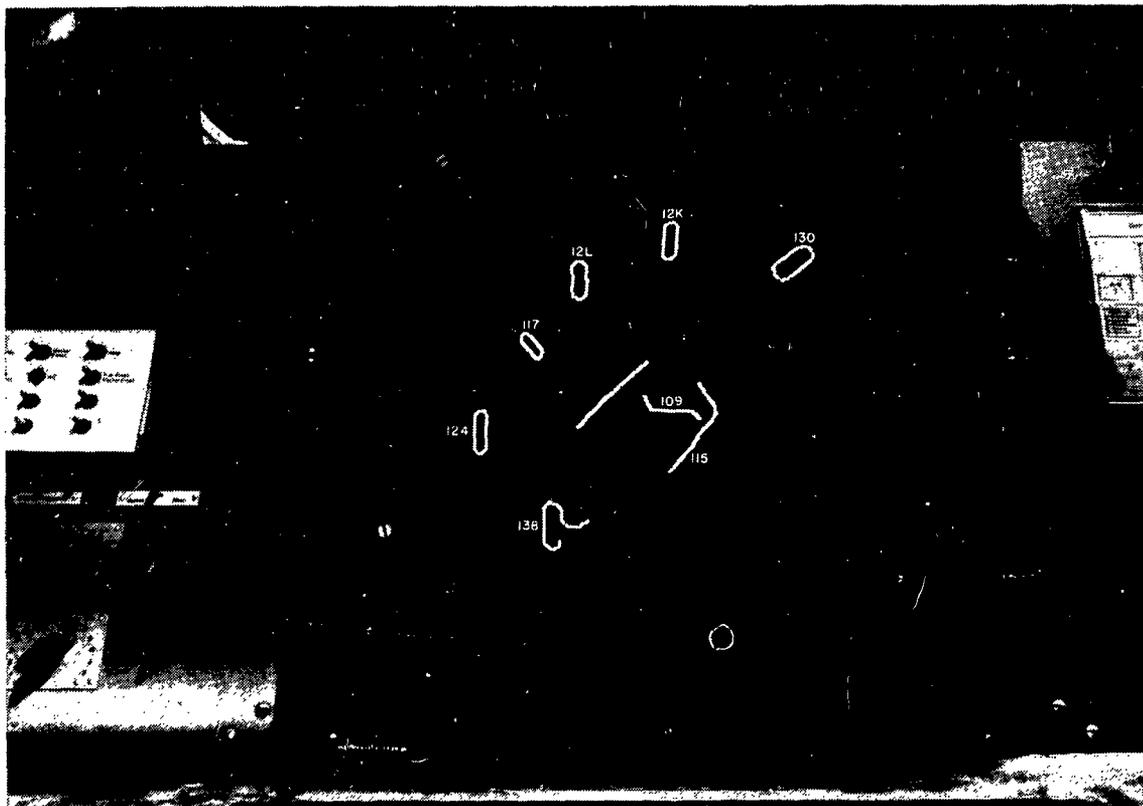


Fig. 31. This exposure covers the time interval from 12 minutes to 16 minutes. Tri-pacer 10J landed at 15 min. 34 sec., 20 seconds after 112 had crossed the outer marker in violation of the sequencing rule. At the end of the exposure, 112 is about half way between the outer marker and the runway, tri-pacer 10J has been vectored NW to increase spacing behind 112 and Cessna 310 (115) is being spaced behind 10J. In the south, jet 138 has been selected for sequencing behind 115 and has just been turned to a heading 040° . All other traffic continues to hold.

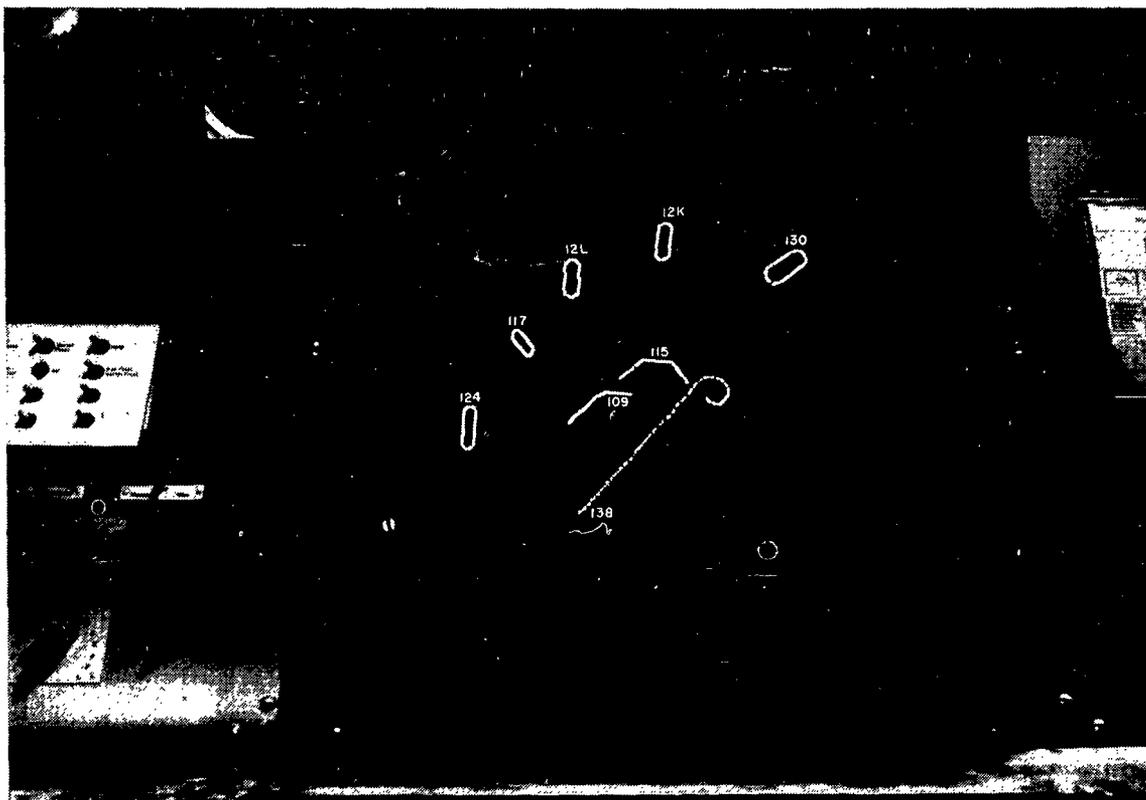


Fig. 32. The exposure covers the time interval from 16 minutes to 20 minutes. Cessna 310 (112) landed early in the exposure at 16 min. 39 sec. At the end of the exposure, tri-pacer 109 is still on the runway and Cessna 115 is on the ILS approaching the outer marker. Jet 138 has been held for 1 minute to increase spacing behind Cessna 115. All other traffic continues to hold.

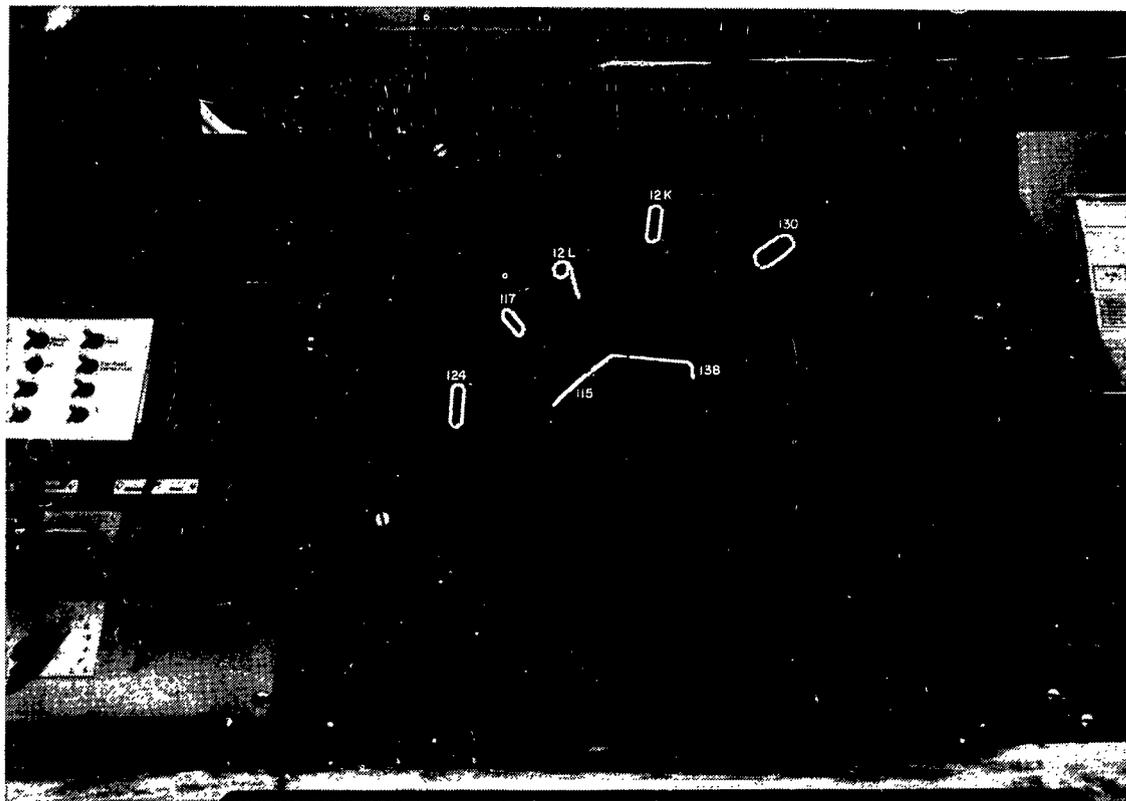


Fig. 33. The exposure covers the time interval from 20 min. to 24 min. Tri-pacer 109 landed early in the photo at 20 min. 43 sec. and Cessna 115 landed at 23 min. 07 sec. At the end of the exposure jet 138 is very near the outer marker and in the NW, electra 12L has been vectored southeast to follow jet 138 in the landing sequence.



Fig. 34. The time interval is from 24 min. to 28 min. Jet 138 landed at 26 min. 13 sec. and electra 12L has passed the outer marker at the end of the exposure. In the west, Cessna 310 (117) has been vectored in to follow 12L and electra 124 has been vectored NE for better position for terminal sequencing. In the northeast, jet 130 has been vectored SW also for better position.

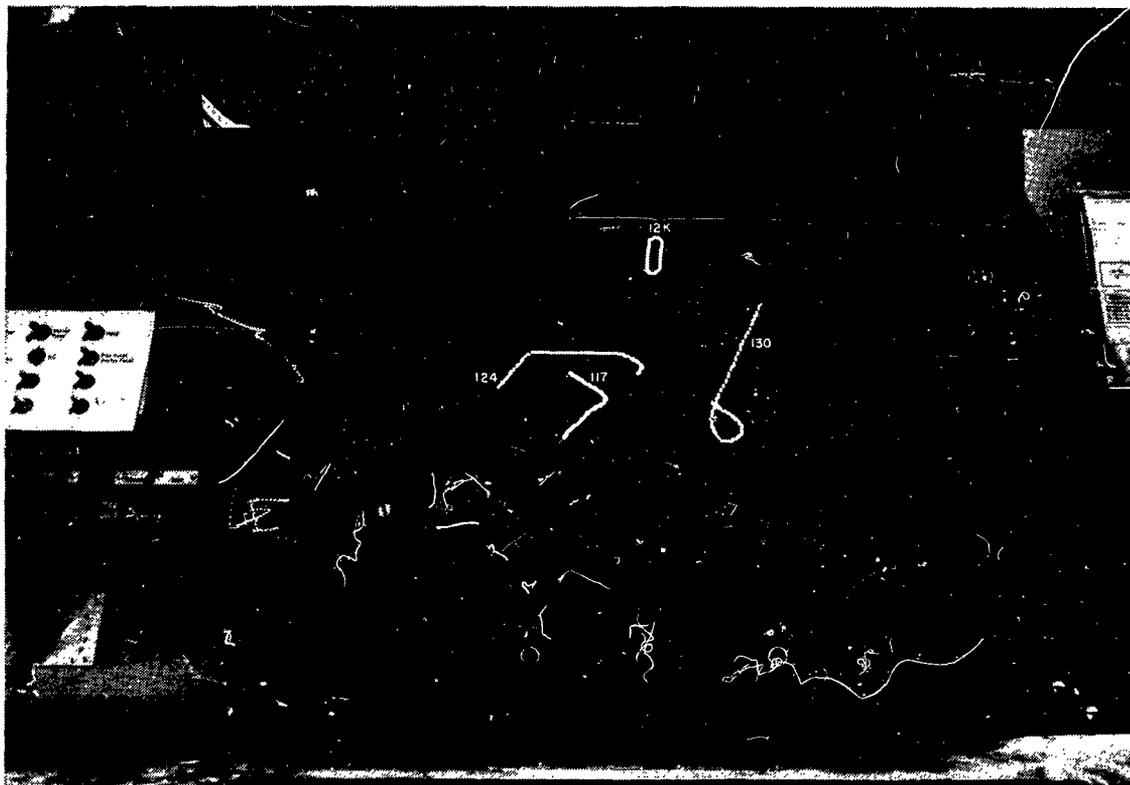


Fig. 35. The time interval is from 28 min. to 32 min. Electra 12L landed at 29 min. 04 sec. and at the end of the exposure, Cessna 117 is just inside the outer marker. From the west, electra 124 has been vectored in to follow 117 in the landing sequence. Jet 130 has been held in the east for 1 minute and at the end of the exposure, is being vectored to follow electra 124 in the landing sequence.

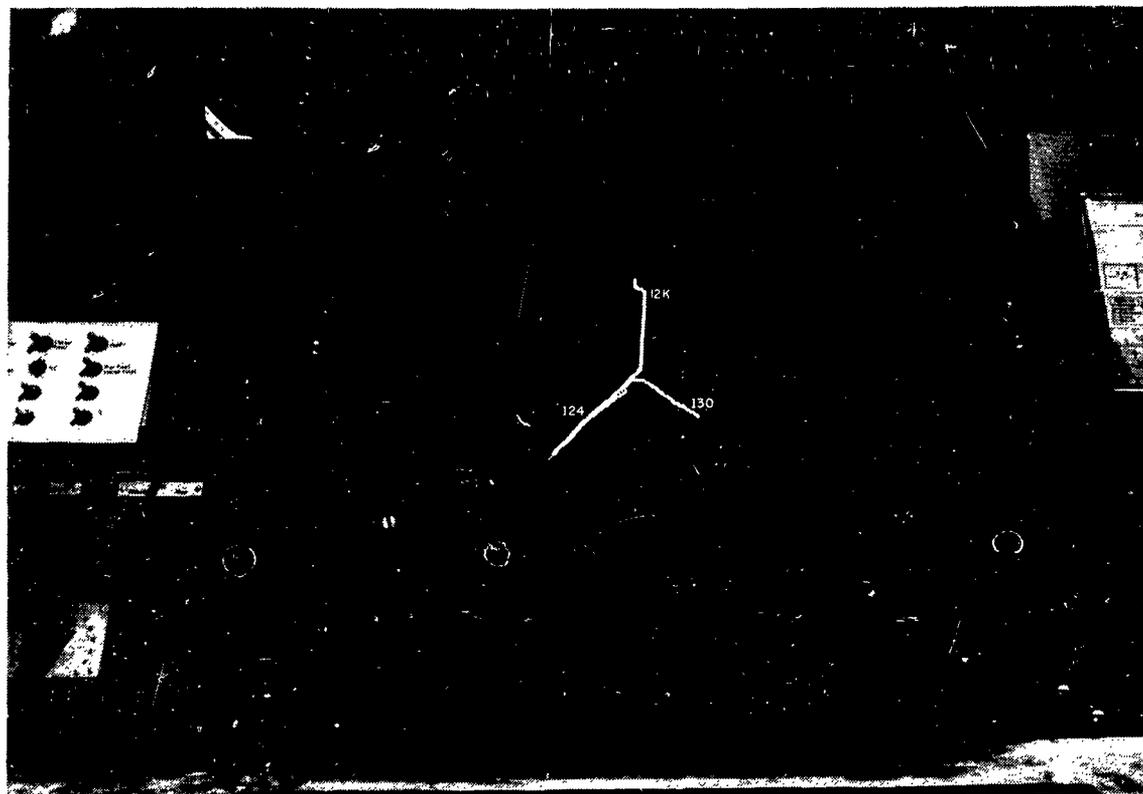


Fig. 36. This exposure covers the time interval from 32 min. to 42 min. 29 sec. at which time electra 12K landed. The landing sequence covered in the photo is electra 124 at 35 min. 36 sec., jet 130 at 38 min. 31 sec. followed by electra 12K at 32 min. 42 sec.

DISTRIBUTION LIST AS OF NOVEMBER 7, 1962

- | | | |
|---|--|--|
| <p>1 Director
Air University Library
Maxwell Air Force Base, Alabama
Attn: CR-4803a</p> | <p>1 Director
Naval Research Laboratory
Washington 25, D.C.
Attn: Code 5140</p> | <p>1 Mailing Zone C-57
Bell Aerosystems Company
P. O. Box 1
Buffalo 5, New York
Attn: Technical Library</p> |
| <p>1 Redstone Scientific Information Center
U.S. Army Missile Command
Redstone Arsenal, Alabama</p> | <p>1 Department of the Navy
Office of Naval Research
Washington 25, D.C.
Attn: Code 437</p> | <p>1 Cornell Aeronautical Laboratory, Inc.
4655 Genesee Street
Buffalo 11, New York
Attn: J. P. Desmond, Librarian</p> |
| <p>2 Hughes Aircraft Company
Florence and Teale
Culver City, California
Attn: N. E. Devereux
Technical Document Center</p> | <p>1 Dr. H. Wallace Sinaiko
Institute for Defense Analysis
Research & Engineering Support Division
1666 Connecticut Ave., N.W.
Washington 9, D.C.</p> | <p>1 Sperry Gyroscope Company
Marine Division Library
155 Glen Cove Road
Glen Cove, L.I., New York
Attn: Mrs. Barbara Judd</p> |
| <p>3 Autonetics
3150 East Imperial Highway
Downey, California
Attn: Tech. Library, 3041-11</p> | <p>1 Data Processing Systems Division
National Bureau of Standards
Conn. at Van Ness
Room 239, Bldg. 10
Washington 25, D.C.
Attn: A.K. Enslow</p> | <p>1 Rome Air Development Center
Griffis Air Force Base, New York
Attn: Documents Library
RAMLD</p> |
| <p>1 Dr. Arnold T. Nordstreck
General Motors Corporation
Defense Research Laboratories
6717 Souther Avenue
Goleta, California</p> | <p>1 Exchange and Gift Division
The Library of Congress
Washington 25, D.C.</p> | <p>1 General Electric Company
Advanced Electronics Center
Cornell University
Ithaca, New York
Attn: F. Kuehn, Librarian</p> |
| <p>1 Inspector of Naval Materiel
Los Angeles, California
Transmittal to:
Litton Systems, Inc.
5500 Canoga Avenue
Woodland Hills, California
Attn: Engineering Library</p> | <p>1 NASA Headquarters
Office of Applications
450 Maryland Avenue, S.W.
Washington 25, D.C.
Attn: Mr. A. M. Greg Andrus
Code FC</p> | <p>1 Library
Eight Military Electronics Department
General Electric Company
Building No. 28-501
Schenectady 5, New York</p> |
| <p>1 Sylvania Electronic Systems - Wes
Electronic Defense Laboratories
P. O. Box 205
Mountain View, California
Attn: Documents Center</p> | <p>1 AFOSG (PCAP)
Edlin Air Force Base
Florida</p> | <p>3 Commanding Officer
U.S. Army Research Office (Durham)
Attn: CRD-4417, Mr. Ush
Box CM, Duke Station
Durham, North Carolina</p> |
| <p>1 Varian Associates
511 Hansen Way
Palo Alto, California
Attn: Dr. Ira Weissman</p> | <p>1 Martin Company
P. O. Box 5537
Orlando, Florida
Attn: Engineering Library
MP-10</p> | <p>1 Goodyear Aircraft Corporation
Fort Project MX 778
Albion 15, Ohio</p> |
| <p>1 Huston Denlow
Library Supervisor
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, California</p> | <p>1 Commanding Officer
Office of Naval Research, Chicago Branch
John Crerar Library Building
10th Floor, 85 East Randolph Street
Chicago 1 Illinois</p> | <p>1 Battelle-DEFENDER
Battelle Memorial Institute
515 King Avenue
Columbus 1, Ohio</p> |
| <p>1 Space Technology Labs, Inc.
One Space Park
Redondo Beach, California
Attn: Acquisitions Group
STL Technical Library</p> | <p>1 Librarian
School of Electrical Engineering
Purdue University
Lafayette, Indiana</p> | <p>1 ASD (ASNC)
Wright-Patterson Air Force Base
Ohio</p> |
| <p>2 Commanding Officer and Director
U.S. Naval Electronics Laboratory
San Diego 52, California
Attn: Code 2800, C.S. Manning</p> | <p>2 Keats A. Pullen, Jr.
Ballistic Research Laboratories
Aberdeen Proving Ground, Maryland</p> | <p>1 ASD (ASRNG)
Wright-Patterson Air Force Base
Ohio</p> |
| <p>1 Commanding Officer and Director
U.S. Navy Electronics Laboratory
San Diego 52, California
Attn: Library</p> | <p>1 Research Analysts Corporation
6935 Arlington Road
Bethesda 14, Maryland
Attn: Library</p> | <p>1 Commanding Officer (AD-5)
U.S. Naval Air Development Center
Johnsville, Pennsylvania
Attn: NADC Library</p> |
| <p>1 Office of Naval Research Branch Office
1000 Geary Street
San Francisco, California</p> | <p>1 Scientific & Technical Information Facility
P. O. Box 2700
Bethesda, Maryland
Attn: NASA Representative (S-AK/DL)</p> | <p>2 Commanding Officer
Frankford Arsenal
Philadelphia 37, Pennsylvania
Attn: SMUTA-1300</p> |
| <p>1 Stanford Electronics Laboratories
Stanford University
Stanford, California
Attn: SEL Documents Librarian</p> | <p>1 Commander
Air Force Cambridge Research Laboratories
Laurence G. Hanscom Field
Bedford, Massachusetts
Attn: CRXL</p> | <p>1 General Atomics Corporation
1075 Delfaven Street
West Conshohocken, Pennsylvania
Attn: Miss D. M. Keener
Librarian</p> |
| <p>1 AFST - SC
Headquarters, USAF
Washington 25, D. C.</p> | <p>1 Research Laboratory of Electronics
Massachusetts Institute of Technology
Cambridge 39, Massachusetts
Attn: Document Room, 26-327</p> | <p>1 H. E. Cochran
Oak Ridge National Laboratory
P. O. Box 2
Oak Ridge, Tennessee</p> |
| <p>1 Director of Science and Technology
Headquarters, USAF
Washington 25, D. C.
Attn: AFST-EL/OU</p> | <p>1 Lincoln Laboratory
Massachusetts Institute of Technology
P. O. Box 73
Lexington 73, Massachusetts
Attn: Library, A-082</p> | <p>1 President
U.S. Air Force Defense Board
Fort Bliss, Texas</p> |
| <p>1 Headquarters, R & T Division
Bolling Air Force Base
Washington 25, D. C.
Attn: RTHR</p> | <p>1 Sylvania Electric Products Inc.
Electronic Systems
Waltham Labs. Library
180 First Avenue
Waltham 54, Massachusetts</p> | <p>1 U.S. Air Force Security Service
San Antonio, Texas
Attn: ODC-R</p> |
| <p>1 Headquarters, U.S. Army Materiel Command
Research Division, R & D Directorate
Washington 25, D. C.
Attn: Physics & Electronics Branch
Electronics Section</p> | <p>1 Minneapolis-Honeywell Regulator Co.
Aeronautical Division
2600 Ridgeway Road
Minneapolis 13, Minnesota
Attn: Mr. D. F. Elwell
Main Station: 625</p> | <p>1 ASTIA Technical Library AFL 2824
Arlington Hall Station
Arlington 12, Virginia
Attn: TISL</p> |
| <p>1 Commanding Officer
Diamond Ordnance Fuze Laboratories
Washington 25, D. C.
Attn: Librarian, Room 211, Bldg. 92</p> | <p>1 Inspector of Naval Materiel
Bureau of Ships Technical Representative
1902 West Minnesota Avenue
St. Paul 4, Minnesota</p> | <p>1 U.S. Naval Weapons Laboratory
Computation and Analysis Laboratory
Delhipton, Virginia
Attn: Mr. Ralph A. Niemann</p> |
| <p>1 Operations Evaluation Group
Office of the CNO (OpO3EG)
Navy Department
Washington 25, D. C.</p> | <p>20 Activity Supply Officer
Building 2504
Charles Wood Area
Fort Monmouth, New Jersey
Marked: For Office of Engineering Operations
Order No. 39578-PM-60-91-91</p> | <p>2 Army Materiel Command
Research Division
R & D Directorate
Bldg. T-7
Gravelly Point, Virginia</p> |
| <p>1 Chief of Naval Operations
Tech. Analysis & Advisory Group (OP-071)
Pentagon
Washington 25, D. C.</p> | <p>1 Radio Corporation of America
RCA Laboratories
David Sarnoff Research Center
Princeton, New Jersey
Attn: Library</p> | |
| <p>1 Commanding Officer & Director
David W. Taylor Model Basin
Navy Department
Washington 7, D. C.
Attn: Code 142, Library</p> | <p>1 Dr. J. H. Frank
861 Washington Avenue
Westwood, New Jersey</p> | |
| <p>1 Bureau of Ships
Department of the Navy
Washington 25, D. C.
Attn: Code 686</p> | <p>1 Mr. A. A. Lundstrom
Bell Telephone Laboratories
Room 2E-127
Whippany Road
Whippany, New Jersey</p> | |
| <p>1 Bureau of Ships
Navy Department
Washington 25, D. C.
Attn: Code 732</p> | <p>1 AFMDC (MDSGP/Capt. Wright)
Holloman Air Force Base
New Mexico</p> | |
| <p>1 Technical Library, DL-3
Bureau of Naval Weapons
Department of the Navy
Washington 25, D. C.</p> | | |