

82.3

FTL REPORT R82-3

MIT

**FTL COPY, DON'T REMOVE
33-412, MIT 02139**

DECISION SUPPORT SYSTEMS FOR
AUTOMATED TERMINAL AREA
AIR TRAFFIC CONTROL

BY

JOHN DEMETRIOS PARARAS

SEPTEMBER 1982

DEPARTMENT
OF
AERONAUTICS
&
ASTRONAUTICS

FLIGHT TRANSPORTATION
LABORATORY
Cambridge, Mass. 02139

FTL REPORT R82-3

DECISION SUPPORT SYSTEMS FOR AUTOMATED TERMINAL AREA
AIR TRAFFIC CONTROL

John Demetrios Pararas

September 1982

- blank page -

ABSTRACT

This work studies the automation of the terminal area Air Traffic Management and Control (ATM/C) system. The ATM/C decision-making process is analyzed and broken down into a number of "automation functions". Each of these functions is described with particular emphasis on its role in the overall system and its interactions with the other ATM/C automation functions. Runway Scheduling and Traffic Flight Plan Generation are identified as the two functions with the greatest potential for providing efficiency improvements over the current terminal area ATC system and are studied in detail.

A very general Mixed Integer Linear Programming (MILP) formulation of the Runway Scheduling problem is developed. Less general formulations and algorithms which have appeared in the literature are reviewed and evaluated. A heuristic algorithm is developed. The algorithm is based on the work of Dear and adopts the Maximum Position Shifting methodology proposed by him [DEA 76]. It extends Dear's work in several ways: (1) it is applicable to multiple runway configurations, (2) it is designed to operate in a real-time simulation environment, and (3) it is designed to accept arbitrary constraints imposed by the ATM/C controller.

The methodology for generating flight plans is developed. Flight plans are based on a specified route structure. They are 4-dimensional and conflict-free. To allow maximum runway scheduling flexibility, a specific route structure is proposed. It is designed to allow easy modification of flight plans to adapt to the dynamically changing schedule.

To allow algorithmic development and testing of this (as well as other) ATM/C automation concepts, a real-time terminal area simulation facility (called TASIM) is designed and implemented. The facility has a number of characteristics which make it a good general purpose tool for terminal area ATM/C research:

- (1) Highly modular design which allows addition, removal and modification of functions with relative ease.
- (2) Realistic modelling of the aircraft dynamics of motion and the aircraft guidance system. Errors introduced by the navigation equipment (onboard and on the ground) and by the surveillance radars are also modelled.

- (3) Capability to simulate multiple controller positions
- (4) Flexible controller interface which allows easy implementation of alternative displays and alternative protocols for man-machine interaction.

The simulation is fully operational in the conventional (manual) ATC mode. In addition, it is currently interfaced with an implementation of the runway scheduling heuristic, and with a special purpose final vectoring display designed to aid the controller in precisely timing the delivery of landing aircraft at the outer marker.

Acknowledgement

Professor Robert Simpson has been a source of advice and inspiration throughout the course of this work. He has been the teacher who introduced me to the field of Air Traffic Control, and patiently relayed his vast knowledge on the subject over the last few years. He has been the advisor who provided guidance when I most needed it, and at the same time allowed me the freedom to pursue my own ideas. Working under his supervision has been an extremely rewarding and pleasing experience. For this I am deeply grateful.

Professor Amedeo Odoni has shaped my entire graduate career at MIT. My admiration for his rare combination of academic achievement, teaching ability, and deep love and respect for people, was immediate and has kept growing over the years. I feel privileged to have known him and to be able to call him my friend.

Professor Thomas Magnanti sparked and has kept alive my interest in mathematical programming and optimization. His inspired teaching and valuable guidance during my years at MIT are deeply appreciated.

I would also like to express my thanks to Gabriel Handler who introduced me to the field of flight transportation and is thus indirectly responsible for bringing me among the outstanding group of people which comprise the Flight Transportation Laboratory at MIT.

Finally, my wife Jan and my son Dimitri provided the stability which saw me through the difficult moments and multiplied my feeling of accomplishment during my successes.

The research for this project was performed at the MIT Flight Transportation Laboratory. Funding for the development effort was provided by NASA Langley Research Center through the Joint University Program for Air Transportation Research. (NGR-36-009-017)

- blank page -

TABLE OF CONTENTS

	Page
CHAPTER 1: INTRODUCTION	14
1.1 Background and Motivation	14
1.2 Overview of Previous Research	19
1.3 Document Summary	25
CHAPTER 2: FUNCTIONAL DESCRIPTION OF AN AUTOMATED TERMINAL AREA ATM/C SYSTEM	 29
2.1 Introduction	29
2.2 The Generic Terminal Area ATM/C System	30
2.3 The ATM/C Flight Plan Generator	38
2.3.1 Nominal Flight Plan Generator	40
2.3.2 Runway Scheduler	41
2.3.3 Traffic Flight Plan Generator	44
2.4 ATM/C Command Processor	45
2.5 The ATM/C Traffic Monitor	49
2.5.1 Conformance Detection and Resolution	49
2.5.2 Hazard Detection	50

TABLE OF CONTENTS

(continued)

	Page
CHAPTER 2 (continued)	
2.5.3 Hazard Resolution Path Generator	52
2.6 Traffic and CTID Displays	53
2.7 The ATM/C Controller's Role	55
CHAPTER 3: A SIMULATION TESTBED FOR THE DEVELOPMENT OF ATM/C AUTOMATION SOFTWARE	57
3.1 Overview	57
3.2 The Hardware Environment	58
3.3 Major System Components and Interfaces	60
3.4 The Simulation Monitor Process	64
3.4.1 Data Base Initialization	64
3.4.2 Interprocess Communication	68
3.4.3 Subordinate Processes and Execution Control	70
3.4.4 Initialization of Terminal I/O	73
3.4.5 Timer Loop Alert	74
3.4.6 Delay Timer Alert	76
3.4.7 Screen Update Timer Alert	76
3.4.8 Mailbox Message Processor	78

TABLE OF CONTENTS

(continued)

	Page
CHAPTER 3 (continued)	
3.4.9 Terminal Input Processing	79
3.5 The Position Generator Process	81
3.5.1 Mailbox Message Processor	82
3.5.2 Traffic Generation Model	83
3.5.3 Aircraft Status Update and Handoff Processing	86
3.5.4 Aircraft Dynamics	87
3.5.5 Air Data System Update	88
3.5.6 Aircraft Navigation System	88
3.5.7 Surveillance System	89
3.6 The ATM/C Controller Process	91
3.6.1 Mailbox Message Processor	93
3.6.2 Command Activation	94
3.6.3 Traffic Display Driver	97
3.6.4 Input Editor	102
3.6.5 Input Processor	102
3.6.6 CTID Update	104
3.7 Implementation Status	106

TABLE OF CONTENTS

(continued)

	Page
CHAPTER 4: THE RUNWAY SCHEDULING PROBLEM	107
4.1 Introduction	107
4.2 Definitions	109
4.3 Formulation of the Dynamic Runway Scheduling Problem	113
4.4 Measures of Runway Efficiency	117
4.4.1 Runway Capacity	118
4.4.2 Aircraft Delays	120
4.4.3 User Cost Distribution: The CPS Methodology	125
4.5 The Complexity of the Runway Scheduling Problem	128
4.6 Variations of the Runway Scheduling Problem	134
4.6.1 Scheduling Landings on a Single Runway (MPS=infinity)	135
4.6.2 Scheduling Landings on a Single Runway (MPS < $N_a - 1$)	138
4.6.3 Scheduling Landings on Independent Runways (MPS < $N_a - 1$)	143
4.7 Heuristic Versus Exact Algorithms	145
4.8 A Heuristic Algorithm for the Scheduling Problem	150
4.8.1 The Initialization Stage	151

TABLE OF CONTENTS

(continued)

	Page
CHAPTER 4 (continued)	
4.8.2 The Feasibility Stage	154
4.8.3 The Optimization Stage	156
4.9 Implementation Status	162
4.10 Coordination Among Terminal Area Control Sectors	163
4.11 Computer Aided Vectoring for Approach Spacing	166
CHAPTER 5: A METHODOLOGY FOR TRAFFIC FLIGHT PLAN GENERATION	171
5.1 Introduction	171
5.2 Flight Path Structures	175
5.3 Ground Track Selection and Speed Control	184
5.4 Altitude Selection	188
5.5 Conflict Identification and Resolution	189
5.6 The Base and Final Approach Phases	192
5.7 Aircraft Guidance	207
5.7.1 Recovery from Errors in the Execution of Turns	207
5.7.2 Speed Control	211

TABLE OF CONTENTS

(continued)

	Page
CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS	215
6.1 Summary	215
6.2 Recommendations for Future Research	219
APPENDICES	
A Interarrival Dynamics	222
B The Kinematics of the Intercept Vector	229
REFERENCES	233

LIST OF FIGURES

	Page
2-1 Overall Schematic Diagram of the Terminal Area ATM/C System	31
2-2 The Major Elements of the Automated Ground Control System	35
2-3 Functional Block Diagram of the Automated Ground Control System	39
3-1 Current TASIM Hardware Diagram	59
3-2 TASIM Software Processes	61
3-3 Top Level Flowchart of the Simulation Monitor Process	65
3-4 Timer Loop Alert Processing Detail	75
3-5 Delay Timer Alert Processing Detail	77
3-6 EXIT Message Processing Detail	80
3-7 Functional Diagram of the ATM/C Controller Process	92
3-8 Traffic Display Layout	99
3-9 Controller Tabular Information Display Layout	105
4-1 Runway Scheduling Heuristic (Initialization Stage)	152
4-2 Runway Scheduling Heuristic (Feasibility and Optimization Stages)	157
4-3 Final Vectoring Display	168
5-1 Typical Terminal Control Area Layout	176
5-2 Typical Nominal Approach Flight Plan	177

LIST OF FIGURES

(continued)

	Page
5-3 Example of an Approach Path Structure	182
5-4 Typical Flight Plan for the Base and Final Approach Phases	193
5-5 Horizontal Separations During the Final Approach Phase	197
5-6 The Kinematics of Aircraft Turns	208
5-7 Recovery from Early Speed Reduction in the Downwind Phase	212
A-1 Minimum Interarrival Separation (Overtaking Case)	225
A-2 Minimum Interarrival Separation (Opening Case)	225
B-1 The Kinematics of the Intercept Vector	230

LIST OF TABLES

	Page
3-1 ATC Radar Beacon System Parameters	90
5-1 Effect of Aircraft Speed on Horizontal Separations and on the Timing of Descent	202
A-1 Minimum Interarrival Separations at the Runway	226
A-2 A comparison of Landing Sequences	228

- blank page -

CHAPTER 1

INTRODUCTION

1.1 Background and Motivation

Over the last twenty years advances in computer hardware have had a tremendous impact on every aspect of life. The civil air transportation industry, due to its close ties with the defense industry and the space program, has traditionally been among the frontrunners in the use of advanced computer technology. Today, computers play a crucial role in practically every aspect of air transportation. Starting from computer assisted design and manufacturing of aircraft and going all the way to autopilots and automatic flight control systems, computers perform tasks of increasing number and complexity.

The Air Traffic Control (ATC) system in the United States has experienced the same impact. The effort towards air traffic control automation started over twenty years ago with the design and implementation of the current (third) generation Air Traffic Control system. TRACONs (Terminal Radar Control) and ARTCCs (Air Route Traffic Control Centers) are the current ATC system's terminal area and enroute facilities respectively. Together they provide a nationwide network of computers which store, maintain, and distribute information on all IFR flights.

In the present ATC system, automation has been accomplished in data collection and processing. Flight plan and surveillance data are automatically collected and distributed to appropriate stations. Radar returns are automatically correlated and filtered so that the controller has better position information than was previously available. Identity and altitude information, supplied by beacon radar for suitably equipped aircraft, is associated with radar returns. All the available data are used to generate clear alphanumeric traffic displays for the air traffic controller.

Generally, flight processing automation has improved the productivity of the air traffic controller by relieving him of many tasks which he had to perform manually up to then. In addition, it has enhanced the exchange of information among ATC facilities thus allowing better coordination. Finally, it has provided controllers with up to date information on current and future flight plans and clearances for IFR flights.

Further improvement of the current system's data processing capabilities will be achieved through the Electronic Tabular Display Subsystem (ETABS). The prototype software for ETABS is currently in its late stages of development under a Federal Aviation Administration (FAA) research and development program. ETABS will replace the "flight strips" that are currently used by the controller. In addition it is designed to enhance inter-controller communication and provide more flexible data entry capabilities than are currently available.

It is generally accepted today that automation in data processing alone is not sufficient to provide the further improvements in the capacity of the ATM/C system required by increasing traffic. This is particularly true in the terminal area which is usually the bottleneck of the entire ATM/C system. Indeed the automation in data processing achieved to date has not provided any increase in the capacity of the facilities comprising the ATM/C system. Instead, it has decreased the controller's workload per aircraft handled so that today, in the terminal area ATM/C system at least, the human element is not the limiting factor determining the system's capacity.¹ It follows that increases in the system's capacity will be achieved only through increased capacity of the airport and the ATC facilities up to the point when the capacity of these elements becomes comparable to that of the controller. Since it is very unlikely that physical expansion of the country's major airports will be feasible in the future, the increases in capacity have to be the result of more efficient use of available resources.

This realization has resulted in efforts to go beyond the data processing automation and design computer software which take part in decision-making for air traffic control. The most important of these efforts are reviewed in section 1.2.

(1) The recent air traffic controllers' strike and subsequent firing has of course changed that situation but the effects are hopefully transient.

The work described here was motivated by the following premises which to some extent deviate from the prevailing trends in current research on the subject of ATC automation:

1. The high traffic density in the terminal area gives high priority to the problems of automating ATC decision-making there. Once satisfactory solutions have been found for the terminal area airspace, the development of an automated system for the enroute airspace should be much simpler. In addition since the terminal area airspace is responsible for a large percentage of the delays experienced in air transportation, any improvements in system operation that can be achieved through automation there will offer much larger benefits than those that can be achieved through automation of the enroute ATC system.
2. Decision automation should result in a true reduction of delays experienced by aircraft. Even though automated spacing of landing aircraft alone will result in some reduction of delays by avoiding excessive gaps between aircraft during the final approach phase, we believe optimal scheduling of all runway operations to have much greater potential for delay reduction and we therefore consider it to be the primary ATC automation function. We propose, therefore, to view the problem not only as one of

traffic control but also one of traffic management. In order to stress this view we will, hereafter, use Hsin's term Air Traffic Management and Control (ATM/C) system instead of the more commonly used term "Air Traffic Control". [HSI 76]

3. The proper man-machine relationship corresponds to "master-slave" with the human master accepting responsibility for all decision making. He should be presented with recommendations by the machine, be able to obtain auxiliary information which supports the decision to satisfy himself that a correct recommendation has been generated, and most importantly be able to override that recommendation by requesting certain conditions to be met which will cause the machine to generate an alternative recommendation. Thus, we insist that the human controller play an active, dynamic role in decision making, and that no decisions are made without his explicit approval. We reject a passive, monitoring role for the human where he may somehow exercise a veto over machine decision making, and presumably intercede on an exception basis. Instead we see the human as the decision maker and the machine as

a "decision support" system which generates decision alternatives for him.¹

1.2 Overview of Previous Research

The topic of ATM/C automation has received much attention over the last two decades. A number of studies have been conducted and are continuing (see for example [ATH 71], [SAR 71], [SCH 73]). We will not attempt to review all of them here but will restrict ourselves to the most important ones from the point of view of results as well as the impact they have had in shaping the course of future work in the area.

Certainly the two most comprehensive studies in terminal area ATM/C automation have been the Metering and Spacing (M&S) program sponsored by the Federal Aviation Administration (FAA) and the development of the Marine Air Traffic Control and Landing System (MATCAL) currently under development by the US Marines.

The FAA's M&S program ([TAL 78], [TAL 80]) has its origins back in the early 60's. Its current version is the third of a series of field test programs. The first two, FASA (Final Approach Spacing for ARTS) and CAAS (Computer Aided Approach Spacing), suffered from a variety of problems (including procedural incompatibility, increased controller workload and in the case of CAAS the lack of a tracker) and thus yielded

(1) The term ATM/C automation will be from here on used to mean automation of ATM/C decision support as contrasted with data processing automation which was discussed earlier.

few conclusive results. Computer simulations using the current version of the M&S system are being currently conducted at the NAFEC facility in Atlantic City.

The primary objective of the M&S system is to increase airport landing capacity by providing more consistent inter-arrival spacing of landing aircraft than is now attainable, thus assuring an increase in runway utilization. The current M&S design does not provide for multiple airports or for multiple runway operations at the same airport. Provision for departures is made only through use of normal gaps in the landing stream as well as manually entered requests to lengthen the landing interval by the controller. The landing sequence is determined based on the nominal time of arrival at the runway and resequencing of landings occurs only in cases where an aircraft cannot arrive at the runway or at one of the intermediate waypoints at the assigned time. Forward slippage and subsequent resequencing is also possible in the case of aircraft that are too early at one of their intermediate waypoints.

The current M&S effort marks the first time that an automated flight plan generator has been accomplished. This has been a step in the right direction, namely away from complex algorithms based on optimal control theory (e.g. [SAR 71] and [SCH 73]) and towards simpler and faster path control techniques consistent with today's aircraft navigation capabilities. The objectives and scope of the M&S program has been limited due, apparently, to a decision by the FAA that some

automation system was needed quickly. Though this is certainly true, there is a need for longer term planning of a comprehensive terminal area ATM/C automation system which this work attempts to lay the foundations for.

By focusing on a quickly implementable system, the FAA M&S system has incurred some serious drawbacks two of which have already been mentioned (handling of single runway configurations only, and the fact that departures are only implicitly taken into account). Another drawback in the current M&S work is the lack of conflict resolution. The flight plans generated for arriving aircraft are not checked for future violations of ATC separation standards. The human controller is required to provide altitude separation whenever flight plans for two aircraft are in conflict. A closely related issue is the system's lack of capability to recover and continue performing the required functions after controller intervention.

Finally, strict adherence to the first-come-first-served sequence in scheduling runway operations will unduly reduce the M&S system's efficiency. Optimizing the runway schedule based on the aircraft mix on hand, was considered but was not incorporated in the current M&S system¹ since

(1) The FAA has recently sponsored a study to determine the capacity improvements that can be expected from optimal runway scheduling and revise the decision if satisfactory levels of improvement are found possible.

" ...the actual [capacity] improvement [from optimum scheduling] is expected to be less than 3% ... [and] can only be realized when system load is very high. But to achieve the improvement the sequence will appear abnormal (compared to current ATC practices). The abnormal sequence will tend to increase controller workload (since the system's intent will be obscure) and, under heavy loads any increases in workload or any enigmatic system performance must be judged inappropriate."

[TAL 78]

The 3% quoted in this paper is too low an estimate of the expected improvement in capacity. The work of Dear, [DEA 76], indicates that improvements in the 10 to 15 percent range are achievable. This relatively small capacity improvement can result in dramatic delay reductions when the airport is operating near saturation. It is true however that this improvement can be achieved only by implementing a flight planning algorithm which is flexible enough to accommodate frequent changes in the schedule of aircraft in their initial approach phase.

It is also not clear what is meant in the above statement with regard to the obscurity of the system's intent and the enigmatic system performance. There is nothing enigmatic about changing the sequence of operations in order to achieve better runway efficiency. In fact, final approach controllers today recognize the efficiency gained by sequencing aircraft of similar landing speeds in direct succession, and do it whenever it can be done easily. The optimization of runway schedule causes the same types of groupings to occur.

The Marine Air Traffic Control And Landing System (MATCALs) is being implemented in response to operational requirements to upgrade and automate the terminal air traffic control and all-weather landing control capabilities of Marine Air Traffic Control Squadrons (MATCS). It is intended to be a deployable system, designed to replace existing MATCS equipment. The MATCALs concept will provide significantly improved capabilities through automation and advanced sensors, data links, displays and operator consoles.

MATCALs provides automated surveillance and traffic control throughout the airspace within 60 nautical miles of the airfield. In addition it provides automatic and semi-automatic landing guidance and control under all weather conditions down to zero ceiling for suitably equipped aircraft.

MATCALs will be developed in three stages each with increasing capabilities. The first stage will be completed in the early 80's and its capabilities will be comparable to today's ARTS III system. The second stage will include automatic traffic monitoring and hazard detection algorithms. The third and final stage will be a fully automated ATM/C system including such functions as runway scheduling and flight plan generation. The runway scheduling and flight plan generation algorithms described in this document will be the prototype software for the third stage of the MATCALs system.

The MATCALs effort should provide valuable new insight into the problem of ATM/C automation and, even though it is concerned with military operations, the MATCALs solutions and experience will be of great value in the development of an advanced automated ATM/C system for civil aviation.

In parallel to the M&S system, the FAA is currently developing an automated ATC system for the En Route airspace. This system, called AERA (Automated En-Route Air Traffic Control), will centralize the flight planning for the en route airspace. The concept of AERA is defined in [GOL 81]. AERA will incorporate all the automation functions that have thus far been developed by the FAA such as conflict alert, en route metering, Automatic Traffic Advisory and Resolution Service (ATARS), as well as state-of-the-art communications and display technology such as the mode-S beacon provided by the Discrete Address Beacon System (DABS), and the Electronic Tabular Display Subsystem (ETABS). It will generate and maintain four dimensional conflict-free flight plans for all IFR flights within the planning region. Aircraft characteristics such as true airspeed, optimal climb and descent profiles, etc. will be used to insure that projected flight plans are closely matched to the aircraft capabilities. In addition AERA will provide routine aircraft separation and traffic flow control, as well as clearance generation, delivery and acknowledgement functions.

The AERA concept will improve controller productivity by relieving controllers from many of the routine functions for which they are

currently responsible. In addition it will make more efficient use of enroute airspace by limiting the need for procedural separation of aircraft and by allowing freer movement of traffic capable of area navigation. Finally AERA will coordinate the transition of traffic from the enroute to the terminal area airspace and automatically perform flow control whenever necessary.

The AERA concept is an ambitious undertaking with a long term planning horizon. It is our hope that the FAA will initiate a similar program to develop the terminal area ATC system for the year 2000 and beyond.

1.3 Document Summary

The work described here is part of a continuing research effort at the MIT Flight Transportation Laboratory towards the development of prototype software for an automated terminal area ATM/C system. In designing such a large scale software system, it is critically important to clearly define its operation in terms of a number of functionally distinct but interacting elements. This breakdown is necessary in order to understand the ATM/C system at the conceptual level. Furthermore it is an essential step towards better organization of software into functionally related entities (or modules). Finally, by bringing out the interactions between various system modules it insures that the functional specifications for each module will be compatible with the

operation of the overall system. The functional breakdown developed during the course of this research is presented in chapter 2.

The issue of compatibility among the functional specifications of various system modules is most apparent in the interactions between the man-machine interface and the automation software. Even though we realized at the outset that considerable research needed to be done before a successful design of the man-machine interface were achieved, we soon realized that the algorithmic development had to depend on the interactions with the air traffic controller. This means that human factors affect not only each module's function but also the algorithm used to perform that function. This realization led to the development of a real-time interactive ATM/C simulation facility called TASIM. The development effort for TASIM is presented in chapter 3.

TASIM is currently being modified, under a NASA/Ames contract, to simulate both terminal area and enroute airspace. In addition to the controller stations, it will include pseudo-pilot stations. The new software will provide the air traffic control environment for the Man-Vehicle Systems Research Facility (MVS RF) currently under development by NASA/Ames, [PAR 82]. In addition to the ATC subsystem the MVS RF includes two cockpit simulators and will be used as a testbed for research in cockpit as well as ATM/C automation.

In parallel with the development of TASIM, a real-time heuristic algorithm for automatic runway scheduling was designed and implemented.

The algorithm can handle multiple runway configurations. To date, however, it has only been successfully tested for single runway configurations. Chapter 4 discusses the problem of runway scheduling; various alternative ways to formulate the problem are presented and solution procedures are discussed; the algorithm which has been implemented is described along with particular requirements that are imposed on it by the nature of the system in which it will operate. This chapter also presents the concept for a final approach controller display which is designed to be used in connection with the automated runway scheduling function in order to allow precision delivery of landing aircraft at the runway in the absence of flight plan automation.

The final part of the research addressed the problem of automatic flight plan generation. The general approach has been that first, flight plans must be compatible with conventional navigation capability, and second, flexibility in path stretching and shortening must be preserved to the greatest extent possible at every point along the flight plan in order to provide maximum rescheduling flexibility. Accordingly, flight plans consist of a number of linear legs, and are generated based on a prespecified ground track structure. This structure should be capable of providing a number of alternative paths at each intermediate point. One possible structure has been developed and analyzed. Based on that we develop the methodology for the design of a traffic path generation algorithm which is presented in chapter 5.

Chapter 6 summarizes the results and conclusions of the work and identifies topics for further research and development.

- blank page -

CHAPTER 2

FUNCTIONAL DESCRIPTION OF AN AUTOMATED TERMINAL AREA ATM/C SYSTEM

2.1 Introduction

Any complex system can be described at various levels of aggregation. The system components at each level can themselves be complex systems requiring the same type of description. In this chapter the terminal area ATM/C system will be described and analyzed from this point of view. We will begin with the overall system and subdivide it into a number of components or subsystems. We next focus on the subsystem of primary interest in this research, namely the Ground Control system. Finally each of the elements of the Ground Control system will be broken down into functional modules and its operation will be discussed.

At each level we will focus not only on the function of each of the components but also on the interactions, the exchange of information that is required, as well as the flow of information from one component to the others as a result of specific external events (e.g. the ground control system response to a conformance alert). This chapter, therefore develops the framework for the design of the automated ATM/C system and for the understanding, at a top level, of the requirements and the purpose of each of the automation functions. At the same time

it sets the basis for the organization and design of the automation software, since the functional modules in the final breakdown can be regarded as top level software modules for the ATM/C system.

2.2 The Generic Terminal Area ATM/C System

Air Traffic Management and Control (ATM/C)¹ is a complex, interactive system that can be best modeled as a feedback control system. Its elements can be grouped into six generic subsystems:

Aircraft Control system (A/C)

Air-to-Air Data link system (AA COM)

Automatic Ground Control system (AGCS)

Ground-to-Air Data link system (GA COM)

Air-to-Ground Data Acquisition system (AG COM)

Ground-to-Ground Data link system (GG COM)

An overall block diagram of the terminal area ATM/C system is shown in figure 2-1. The major elements are the Automatic Ground Control (AGCS) and the Aircraft Control (A/C) systems. While each of these are feedback control systems themselves, they are elements of the major control loop for ATM/C. The Aircraft Control systems (or "targets") output their "state" vectors, $P_i(t)$ and $P_j(t)$, which are measured by the Data Acquisition (AG COM) system to serve as primary dynamic input data

(1) The material in section 2.2 is drawn from the work of Hsin [HSI 76] where the reader is referred for an excellent detailed discussion of advanced ATM/C systems.

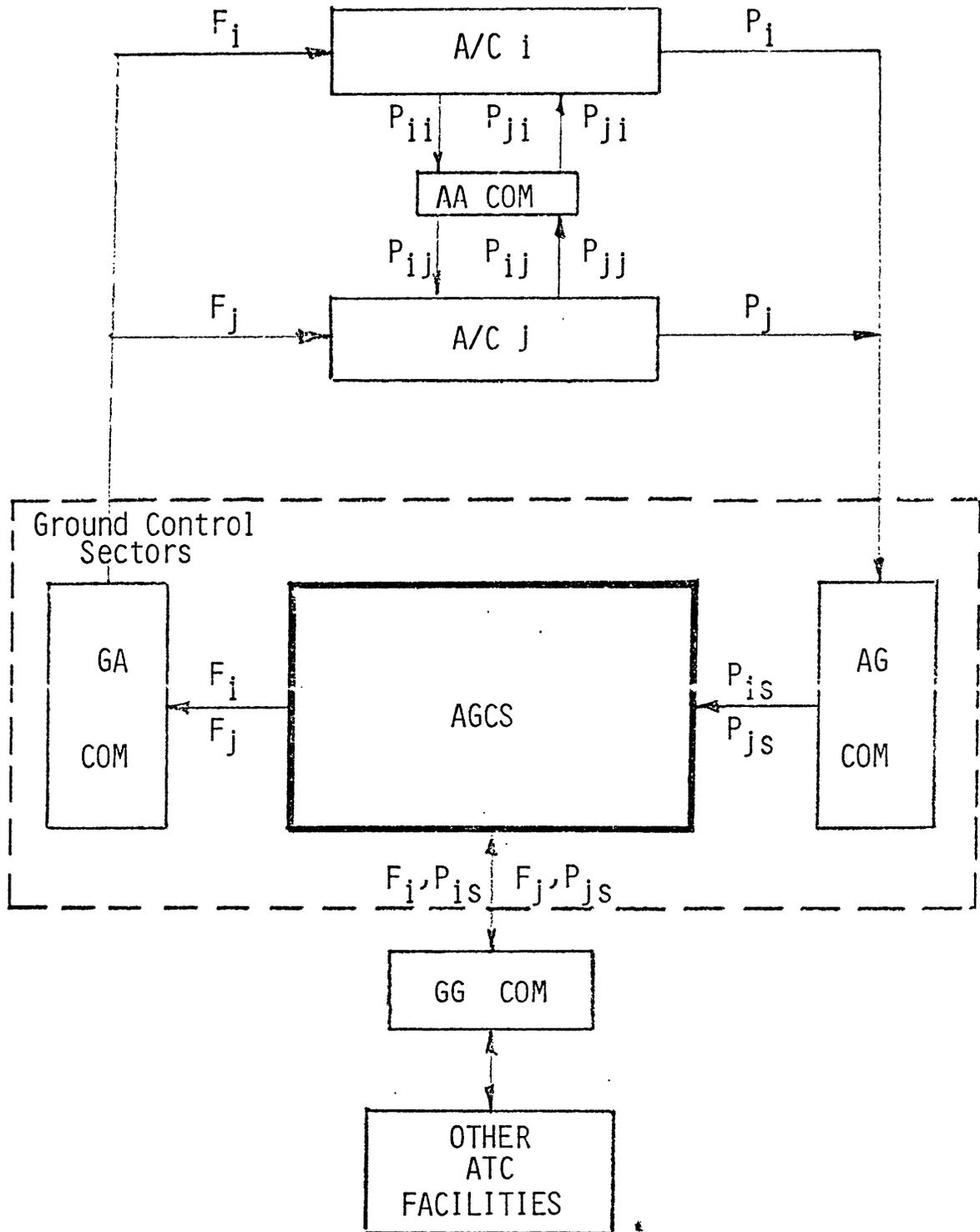


Figure 2-1. Overall Schematic Diagram for the Terminal Area ATM/C System.

to the AGCS. $P_i(t)$ includes information on the past positions of aircraft i . The AGCS processes this information and produces its output, F_i and F_j , which are ATC commands or intended flight plans which describe the future path of the aircraft. These in turn are the primary dynamic inputs to the Aircraft Control systems. The various Data link systems accomplish the required flow of information.

The availability and distribution of information is of key importance in the operation of the ATM/C system. It consists of the current "state", $P_i(t)$, and the "intended" future state or flight plans, F_i , of the aircraft targets in the system. $P_i(t)$ includes the aircraft position and altitude as well as aircraft velocity, accelerations, heading, bank angle, etc. The ATM/C information is not globally available. Aircraft heading, for example, is currently available only to the aircraft control system itself. Due to the variation in its completeness and because of measurement errors the aircraft state vector takes three distinct forms:

1. The actual aircraft state, $P_i(t)$
2. The ground measured aircraft state, $P_{is}(t)$, is estimated information on target i available to sector s of the Ground Control system, and includes the errors introduced by the Air-to-Ground Data Acquisition system.
3. The airborne measured state, $P_{ii}(t)$, is the aircraft state information available to the Aircraft Control system, and

includes the errors introduced by the navigation and guidance subsystems of the A/C. Information on the state of other aircraft in the system can also be available to aircraft i , either in absolute form, $P_{ji}(t)$, or in relative form, $\Delta P_{ji}(t)$. Absolute state vector information may be relayed to other aircraft through the ground to air data link while relative information may be obtained by separation assurance systems onboard aircraft i .

The discrepancy between the ground measured and the airborne measured aircraft state information has far reaching consequences and requires some further elaboration. In addition to their defining difference, (i.e. their topological availability), they also differ in their accuracy, timeliness, and completeness. Their difference in accuracy is obvious since they are measurements produced by two different systems. Also by virtue of the measurement systems involved, the airborne state information is generally instantaneous while ground measurements are made at relatively low sampling rates which depend on the period of revolution of the radar antenna.¹ Finally, while the airborne state information includes a wide variety of measured quantities, (e.g. position, altitude, airspeed, vertical speed, etc.)

(1) There exist, of course, military "tracking" radars which can directly control the direction of the beam thus producing very high sampling rates for particular targets. These radars measure the aircraft position with much better accuracy than conventional radars. It is very unlikely, however, that these will be used for civil air traffic control in the foreseeable future.

today's surveillance systems measure only range and bearing and can receive encoded identity and altitude information from mode-C transponders. Generally therefore, the airborne information on the aircraft state is more complete and in many cases more accurate than the information available to the ground.

The issues of accuracy and availability of aircraft state information are important since these are factors which determine the efficiency and the feasibility of the automated ATM/C system. Sophisticated automation in ATM/C cannot be achieved without high quality surveillance and tracking data. These issues will be discussed again in connection with automated flight plan generation and conformance monitoring.

We can now show the elements within the block for the AGCS shown in figure 2-1. As depicted in figure 2-2, the AGCS has three major functional elements: the Flight Plan Generator (FPG), the Traffic Monitor (ATM), and the Command Processor (ACP).

The FPG is responsible for creating decisions regarding the flow of traffic in the terminal area ATM/C system. It is best described in terms of an optimization process. The state of all controlled aircraft is the input variable of the process. The output will be the ATC commands which define a flight plan¹, F_i , for each controlled aircraft, i . The ATM/C rules and regulations, and the aircraft's dynamic

(1) The term "flight plan" is formally defined later in this section.

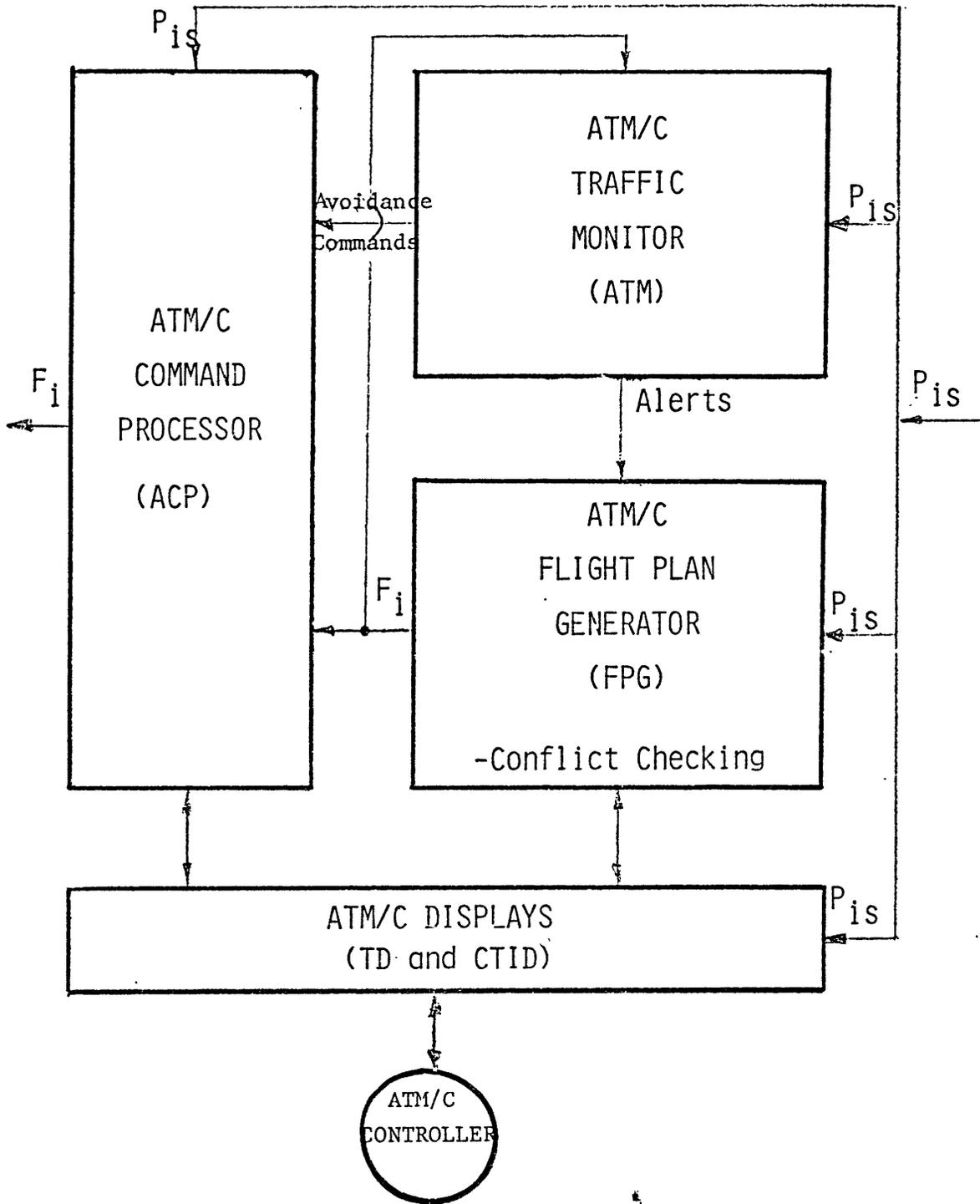


Figure 2-2. The Major Elements of the AGCS.

characteristics (e.g. minimum and maximum airspeeds, etc.) are constraints which determine the set of feasible ground control decisions. Finally, the ATM/C strategy defines the objective function which the FPG is to optimize.

The Traffic Monitor naturally divides into two sub-elements: the Conformance monitor and the Hazard Monitor.

The Conformance Monitor, consisting of the Conformance Detection (CD) and the Conformance Resolution Path Generation (CRPG) functions, is preventive in nature; it monitors aircraft adherence to the ground control decision and generates a conformance alert when deviations from the established flight plans exceed prespecified conformance limits.¹ The Hazard Monitor, consisting of the Hazard Detection (HD) and the Hazard Resolution Path Generation (HRPG) functions, is responsible for enforcement of the ATM/C safety rules; it generates a hazard alert whenever violation of ATM/C standards is imminent. A distinction should be drawn between the Hazard Monitor and the conflict check. "Hazard" implies a perilous current situation in terms of the actual positions of aircraft, whereas "conflict" refers to a possible future situation based on flight plans. In particular, a flight plan is in conflict with another if an aircraft adhering to it within the conformance limits will create a hazard (i.e. violate the ATM/C rules¹) at some future point in

(1) The conformance limits depend primarily on the FPG logic, but the accuracy of the surveillance and the onboard navigation equipment is also a factor.

time. Accordingly the conflict check is an integral part of the FPG and appears in figure 2-2 to indicate that flight plans generated are conflict-free.

The Command Processor also subdivides into two sub-elements: the Command Generator (CG) and the Command Activator (CA). The Command Generator uses flight plan data to determine for each aircraft a set of commands and associated times of initiation, such that if the commands are followed, the aircraft will remain in conformance to its flight plan. The same flight plan may conceivably produce different commands for different aircraft since the aircraft navigation capabilities determine what type of commands the pilot can be expected to accept. The Command Activator is responsible for the timely dispatch of commands to the air traffic controller and (if digital data link is available) to the aircraft.

The only manual activity depicted in figure 2-2 is that of the human controller who controls the overall system performance through real time inputs to the Traffic Display (TD) and the Controller Tabular Information Display (CTID) software. All other elements can have the form of computer algorithms which can operate automatically providing decision support within the framework established by the ATM/C controller. He remains the commander responsible for the system's actions.

Before we discuss the various automation functions in more detail, the following definitions will be useful in describing flight plans. A point W in 3-dimensional space will be called a waypoint. We shall define time-points, $W(t)$, as four dimensional waypoints for which the time dimension is also specified. A flight plan for aircraft i , F_i , is a set of waypoints or time-points defining a path in 3 or 4-dimensional space. We limit our definition of a terminal area flight plan to include only time-points whose time coordinate lies in some interval (t_o, t_f) . Unless stated otherwise, the interval of interest will be the time during which the aircraft is under the control of the terminal area ATM/C controller. For a landing aircraft i , this would be from the time it is handed off to the ATM/C controller to the scheduled time of arrival at the assigned runway n ($STAR_{in}$). For departing aircraft the interval of interest is from the scheduled time of arrival at the runway, i.e. the takeoff time, until the scheduled time of arrival at the exit fix (SXT_i). F_i is feasible if it is consistent with the aircraft's performance characteristics (such as airspeed, climb and descent rates, taxi speeds, etc.).

2.3 The ATM/C Flight Plan Generator

In this section the FPG is described in more detail with particular emphasis on the interactions between the various automation functions that are identified. In figure 2-3 the FPG is shown as consisting of three distinct functions:

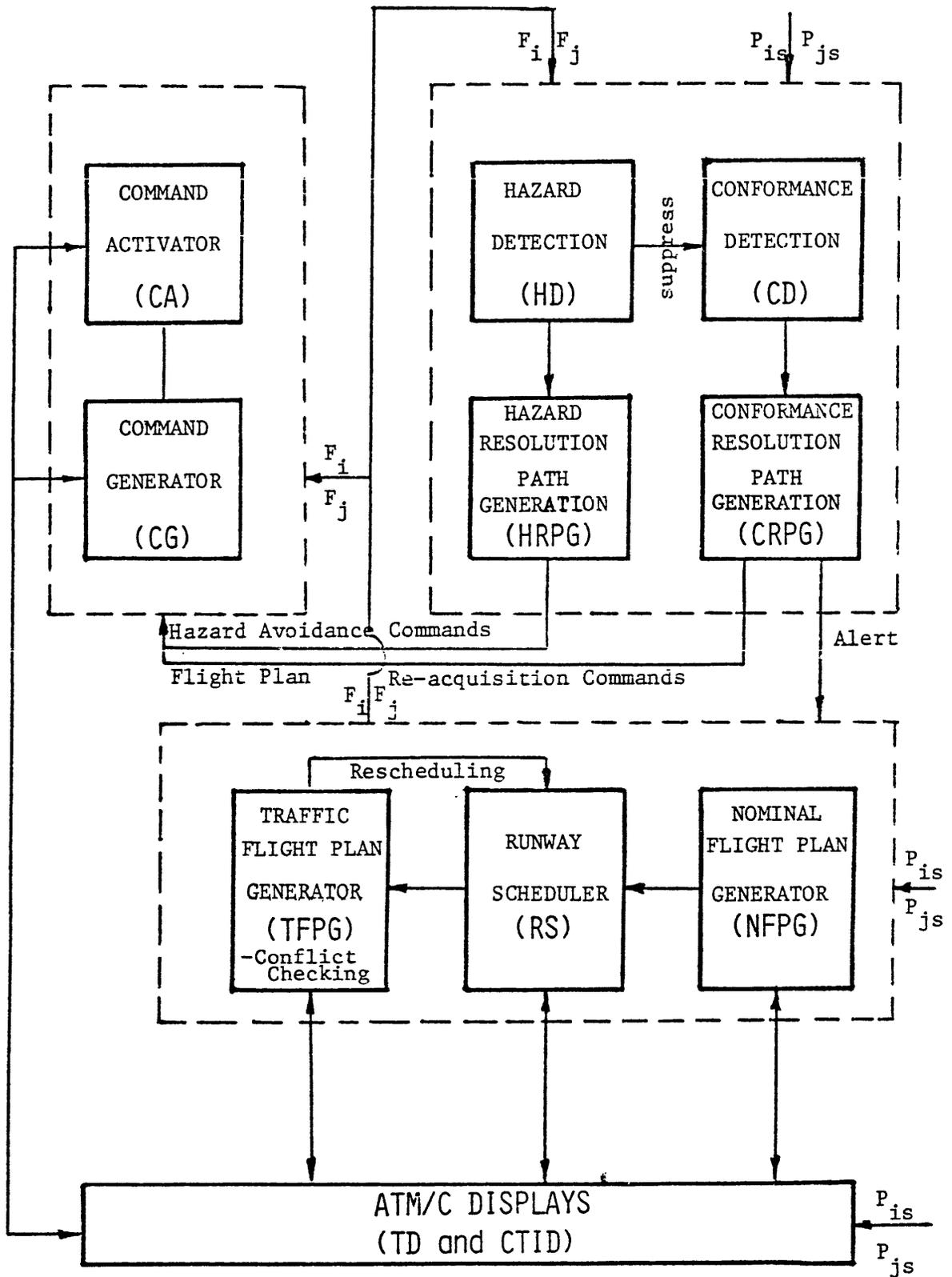


Figure 2-3. Functional Block Diagram of the AGCS.

1. Nominal Flight Plan Generator (NFPG)
2. Runway Scheduler (RS)
3. Traffic Flight Plan Generator (TFPG)

2.3.1 Nominal Flight Plan Generator

Given the present position, destination, and aircraft performance characteristics, the NFPG determines an efficient flight plan, F_0 , that would be assigned to an aircraft in the absence of any other traffic. The flight plan assignment assumes nominal performance characteristics (terminal area speed, descent or climb rates etc.) depending on the aircraft type. Where applicable, standard departure and arrival routes could be used for this purpose. The nominal flight plan establishes earliest times of arrival at various points in the terminal area or at the airport. Of particular interest is the earliest (preferred) time of arrival at the runway threshold (PTAR)¹ which establishes the nominal (first-come-first-served) sequence of operations (NSAR). NSAR is the basis for the sequencing constraints which will be introduced in section 2.3.2

The NFPG is invoked whenever new aircraft enter the system (either for takeoff or for landing) or when the ATM/C controller requests a schedule change (e.g. in the case of a missed approach).

(1) The term "time of arrival at the runway" is used for both landings and departures. For landings it is the time the aircraft crosses the runway threshold, while for departures it corresponds to the time the takeoff roll starts.

2.3.2 Runway Scheduler

Given a runway system consisting of N_r runways, and N_a aircraft, we define a runway schedule S as the set of $STAR_{in}$'s (scheduled times of arrival at the runways) for all aircraft i and runways n in the system.

$$S = \{ STAR_{in} , i=1,2,3,\dots,N_a \quad n=1,2,3,\dots,N_r \}$$

Adoption of a specific runway schedule will be referred to as a scheduling decision. Note that a scheduling decision includes the assignment of runways to aircraft whenever more than one runway is active. If runway n is assigned to aircraft i , $STAR_{im}$ has no meaning for all m other than n . By convention we set $STAR_{im}$ to zero if runway m is not the assigned runway for aircraft i . When two or more runways are active, we may restrict landing and/or takeoff operations to certain runways, perhaps depending on the type of aircraft.

The runway schedule defines the efficiency of the terminal area ATM/C system (i.e. the time interval between successive operations). Generating the schedule thus represents the major optimization effort in the AGCS decision process. Unlike today's manual system where ATM/C objectives and strategies employed to achieve them are only defined in implicit and qualitative terms, the objectives, performance criteria to be optimized, and strategies which are implemented in the automatic system are well defined functions of the $STAR$'s. For example, the performance criterion to be optimized might be average aircraft delay

and the strategy adopted will be a specific runway schedule which accomplishes that. In chapter 4 we discuss in detail alternative performance criteria, their mathematical representation and their effects on runway scheduling algorithms.

The Runway Scheduler is a computer algorithm which solves a well defined optimization problem: Given the nominal sequence at the runways (NSAR), and for each aircraft in the system:

1. its performance characteristics,
2. its Earliest Feasible Time of Arrival, $EFTAR_{in}$, at each active runway n , and
3. its Latest Feasible Time of Arrival, $LFTAR_{in}$, at each active runway n ¹

find a particular schedule S^* which optimizes the performance criterion and satisfies the following constraints:

1. Spacing: the minimum required time separations between all pairs of operations i and j are not violated;

(1) $EFTAR_{in}$ and $LFTAR_{in}$ will depend on the approach routes and the flight planning logic. They will be discussed further in chapters 4 and 5.

2. Sequencing: no aircraft is shifted by more than a prespecified number of positions from its position in the nominal sequence;
3. Scheduling: the scheduled time of arrival at the runway for all landing aircraft should be within the feasible time interval, i.e., for all landing aircraft i assigned to runway n ,

$$EFTAR_{in} \leq STAR_{in} \leq LFTAR_{in}$$

4. Lead Time: The current scheduled time of arrival at the runway cannot change if it is within the prespecified lead time from the current clock time;

Lead time constraints are designed to avoid last minute changes in the schedule. For departing aircraft they allow for taxiing time and smooth operation of the takeoff queue near the runway threshold. These constraints also provide pilots of landing aircraft enough advanced notice of their exact landing time so that adequate preparations for landing can be made. Under normal conditions the STAR for each aircraft stabilizes as its scheduled time approaches. The lead time constraint absolutely ensures that this will always happen at some prespecified time interval before STAR.

2.3.3 Traffic Flight Plan Generator

Given the new runway schedule S^* , the current positions, and the performance characteristics of all the aircraft in the terminal area ATM/C system, the TFPG completes the ground control decision process by generating a new Flight Plan Decision,

$$F^*(t) = \{ F_i^*, i=1,2,3,\dots,N_a \}$$

i.e. 4-dimensional flight plans F_i^* , such that:

1. F_i^* is feasible for aircraft i , $i=1,2,\dots,N_a$
2. minimum airborne separations are satisfied throughout, i.e. the flight plans are conflict-free.
3. a smooth transition between F_i and F_i^* is provided for all $i=1,2,\dots,N_a$
4. The pilot workload from the time of system entry to the time of system exit is kept within reasonable limits.

The flight plans generated here are called traffic flight plans since conflicting traffic is taken into account by the second of the above constraints.

The TFPG is normally invoked after a new scheduling decision has been made due to a new aircraft entering the system or due to a missed approach. It may also be invoked however due to a conformance alert.

There is a possibility that a set of conflict free flight plans cannot be found. In this case the TFPG will invoke the runway scheduler requesting a schedule modification.

2.4 ATM/C Command Processor

This function consists of two sub-functions, the Command Generator (CG) and the Command Activator (CA).

The Command Generator uses flight plan data to generate a set of commands which, if followed, will guide the aircraft along its assigned flight plan. The commands take the form of alphanumeric messages in terminology commonly used today for voice communications. For a specific flight plan the actual commands generated depend on (i) the aircraft's navigational capability, and (ii) the prevailing winds. The aircraft's navigational capability is either be conventional (i.e., VOR/DME or TACAN navigation) or advanced (i.e., 3D or 4D Area Navigation). For conventionally equipped aircraft the set of allowable commands is of the "radar vectoring" type which specify heading, speed and altitude. Aircraft equipped with 3D or 4D Area Navigation (RNAV), can be commanded to track directly to the next waypoint or time-point.

In order to achieve good conformance to 4-dimensional flight plans, speeds and headings should be corrected to account for the estimated winds in the vicinity of the aircraft. Occasionally a segment of a flight plan will coincide with a VOR radial, particularly in the early stages of arrival routes for landing aircraft, and the late stages of departure routes for takeoffs. In that case a command to track a VOR radial would be preferable to a heading command since compensation for wind is performed by the pilot (or autopilot) in tracking. Similarly, RNAV commands will be preferable to heading commands and will be used whenever possible.

Each command message includes at least four additional pieces of information necessary to its further processing:

1. the identity of the aircraft to which the message is directed
2. the time of issuance to the ATM/C controller
3. the time of transmission to the aircraft
4. the acknowledgement of the command by the aircraft

The difference between issuance and transmission time will be such as to give the ATM/C controller the opportunity to study and (if deemed necessary) modify the command, to validate its automatic transmission to

the aircraft at the correct time if digital data link is used for ground to air communications, and to prepare for voice transmission otherwise.

The Command Activator is responsible for the timely dispatching of command messages. The presentation of commands to the ATM/C controller can be achieved in three stages. First, the controller will be able to review all future commands for any or all aircraft under his control. This list of commands can be displayed in the CTID and is updated when flight plans change. The controller may also modify any command at this stage. When the command issuance time has been reached, the command will be moved to the issuance area which may be in the traffic display for easier reference. Additional methods to attract the controller's attention to it (e.g. blinking) may be used. Finally after the transmission to the aircraft has been initiated, the command is moved to a post view area. Thus the controller is reminded of the "active" commands for all aircraft under his control. In this area, the commands may also be flagged when acknowledged by the aircraft crew.

The transmission to the aircraft is initiated when the controller validates the command. Since it can be assumed that the AGCS will be capable of both voice and data link (mode-S) communications, the processing of the validated command by the Command Activator will depend on the aircraft communication capabilities.

The important issue in this respect is that the controller should not be responsible for determining the transmission method. This means

that the Command Activator will determine if the target aircraft has mode-S capability. If so, the command will be moved to the postview area. If not, the command will remain in the issuance area and the controller will realize that he has to transmit the command through voice communication. An interesting option in this regard is the possibility of using voice synthesizers to alleviate the controller workload associated with routine transmission of commands via voice communications. The computer may be able to automatically synthesize and transmit voice commands to all aircraft that do not have mode-S capability.¹

If validation is not made by the specified time of transmission to the aircraft, the Command Activator dispatches an appropriate alert message to the ATM/C controller and no action is taken until the ATM/C controller validates the clearance or initiates another command for the aircraft to follow. Similar processing takes place after the command is transmitted to the aircraft. If a specified time interval has elapsed without acknowledgement of the command by the pilot, appropriate messages are automatically dispatched to the pilot and to the ATM/C controller.

(1) Voice synthesis has tremendous potential in many aspects of ATM/C automation. The technology, however, is still in its infancy and a lot of research is still required with regard to the technological as well as to the human factors aspect before its usefulness can be assessed.

2.5 The ATM/C Traffic Monitor

2.5.1 Conformance Detection and Resolution

This is the primary mode of automatic monitoring for the AGCS. Deviations between the indicated surveillance position of the aircraft and the desired position according to its currently assigned flight plan are monitored. When deviations exceed externally established limits (which may depend on the geographic location of the aircraft with respect to the assigned runway) the aircraft is declared out of conformance with its flight plan and a conformance alert is generated. This will generally cause the TFPG to be invoked.

In some cases it will be reasonable to quickly bring the aircraft back into conformance through immediate "correction" vectoring. This function is performed independent of the TFPG by the Conformance Resolution Path Generator (CRPG). This type of recovery from a conformance alert will be desirable when lateral deviations from the flight plan are detected. Longitudinal deviations, on the other hand, will most likely be the result of discrepancies between the aircraft's airspeed and the nominal airspeed assumed for the purposes of flight planning. In addition to the fact that no aircraft can be expected to fly at exactly its nominal airspeed, such discrepancies will occur due to erroneous estimate of the wind speed and direction particularly after a change in heading. In that case, it will generally be preferable to

modify the flight plan for the aircraft based on the new information (namely the new estimate of the ground speed) available to the TFPG.

2.5.2 Hazard Detection

Hazard Detection is a backup to the primary mode of providing separation assurance, namely through generation of conflict-free flight plans and conformance monitoring. Controlled aircraft must be out of conformance before they can be in hazard.

Traditionally, this function monitors the short term straight line projections (typically of the order of 30 seconds) of aircraft separation from ground terrain as well as the projections of the positions of other aircraft in the system. We will call this the "unassuming" mode (U-mode). This approach has not been very successful, particularly in the terminal area where the proximity of aircraft results in high false alarm rate.

In an automated ATM/C system, flight plan information may be provided to the Hazard Detection function in order to reduce the false alarm rate. We will call this method the "informed" mode (I-mode). There is a certain danger to the I-mode since the Hazard Detection function is primarily responsible for safeguarding against "blunders" either by the air traffic controller or the pilot. The I-mode is not particularly suited to detect such blunders because its assessment of the situation will be biased by the fact that it "knows" what the

aircraft should be doing. It is possible, however, to maintain the same level of performance with respect to blunders and at the same time to increase the reliability of the Hazard Detection function by having both modes operating in parallel.

We can have two levels of hazard alert. The first level is declared when the U-mode declares a hazard but the I-mode determines that it will be resolved according to flight plan information. For example, one of the two aircraft involved is due to initiate a 90 degree turn in the next few seconds. In this case the controller may be warned in order to insure that the expected command is actually in the process of being initiated. No further action is taken unless explicitly requested by the air traffic controller. The second level will be declared when both the U-mode and the I-mode declare a hazard,¹ or when the (internally determined) probability of an actual blunder has reached a specified threshold value (for example, the time that the command in question is to be initiated has been reached). In this case, in addition to the hazard message to the air traffic controller, conformance monitoring for the aircraft will be suspended, and the Hazard Resolution Path Generator will be invoked in order to generate hazard resolution commands. Note that the conformance detection function should be declaring a conformance alert and trying to solve the problem at the same time.

(1) This will generally not be possible if the conflict checking function of the Traffic Flight Plan Generator is operating properly.

A hazard alert status is maintained for the aircraft until it is found not to be in hazard any more. At this time the hazard alert is removed. If there had been a second level alert, conformance monitoring is resumed. Presumably the hazard avoidance path will have placed the aircraft out of conformance with its original flight plan. A normal conformance alert will then be generated causing the CRPG or the TFFG to be invoked in order to solve the problem of returning the ATM/C system in its primary monitoring mode.

2.5.3 Hazard Resolution Path Generator

This function is invoked whenever a second level hazard alert occurs. Two modes of operation are possible depending on the type of ground-to-air communication available.

If voice communication is employed, a recommended avoidance path will be generated and made available to the Command Generator for processing and immediate transmission to the ATM/C controller. The controller then chooses to validate the recommended solution or generate his own. In this mode it may be advantageous to provide the ATM/C controller with more than one alternative.

If a digital ground-to-air data link is available, it may be feasible to generate a single avoidance path and request immediate processing by the Command Generator and transmission to the aircraft. This mode insures faster reaction time to the imminent hazard but

requires a very reliable algorithm for generating hazard avoidance paths. The hazard resolution commands are also transmitted to the ATM/C controller to avoid conflicting actions on his part. He is not however required to take active part in the process of resolving the hazard. He can participate in returning the aircraft into conformance.

The HRRPG is not an absolutely necessary element of the automated system and, at least initially, the ATM/C controller could perform this function manually without any loss in the system performance.

2.6 Traffic and CTID Displays

The human ATM/C controller uses two displays: a normal radar display called the Traffic Display (TD), and an auxiliary display for traffic information called the Controller Tabular Information Display (CTID).

The displays are the output component of the man-machine interface of the AGCS. They can display all pertinent information about the system state (current and future), and performance. The information displayed on either display is controlled by the ATM/C controller via real-time inputs on the associated keyboard or other data entry device. Information nominally displayed on the TD includes:

1. movements of aircraft targets within the radar surveillance area. For every aircraft displayed there is a short alphanumeric block containing the aircraft's

identity, its altitude, estimated ground speed and (for landing aircraft) its current position in the landing sequence;

2. geographical data such as the location of airports, obstructions, etc., as well as the navigational aids located within the surveillance area;
3. commands to be transmitted to controlled aircraft as well as other priority messages such as conformance or hazard alerts.

The CTID will display information that the ATM/C controller will need for reference, but does not require constant cognizance of. Such information includes:

1. All pertinent information on aircraft in the system. This takes two forms: (i) constantly displayed "summary" information including aircraft identification, transponder code, current clearances, etc. and (ii) more detailed information displayed about a particular aircraft when explicitly requested by the ATM/C controller (e.g. full description of the currently assigned flight plan, requested landing speed or takeoff weight for the aircraft etc.);

2. System resource data such as condition of active runways and taxiways, weather data, etc.;
3. System performance data such as the statistics on runway utilization, average delays etc.

2.7 The ATM/C Controller's Role

The ATM/C controller is the human element in the AGCS. He should be the "policy maker" or "chief executive" who determines the strategy and the general framework within which the software is allowed to generate the plan of action down to the very small details.

He controls the decision process in the following ways:

1. overrides the computer generated runway schedule,
2. imposes additional constraints on the runway scheduling process. Such constraints include: requiring that an aircraft land as soon as possible, (as in the case of emergency), require that an aircraft maintain its current position in the sequence, etc.,
3. requests changes in the configuration (e.g., change in the active runways, change in ATM/C strategy, etc.);
4. validates commands for transmission to various aircraft;

5. Edits computer generated commands and initiates their transmission process.

In all cases the ATM/C controller's request is acknowledged and he is informed on the results of the request when such information is necessary to insure that the intended action was taken.

The general philosophy followed here is that the ATM/C controller does not simply monitor the computer performance but remains responsible for safe separation and expeditious flow of traffic, and must be totally in control of the system. The automated system must be a tool to assist the ATM/C controller in carrying out the task safely and expeditiously.

The extent to which automated system will be successfully implemented in the future depends on the success or failure to develop a suitably designed man-machine interface that allows the ATM/C controller to be at all times in control of the overall system and, if desired, capable of exerting such tight control that the computer programs react in a predictable manner from the ATM/C controller's point of view. At the same time, during normal operation a delicate balance should be found between controllability and extensive interference which might completely defeat the purpose of ATM/C automation. Careful examination of the available options and extensive testing of different methods of interactive control will be required before a satisfactory solution to this problem can be reached.

CHAPTER 3

A SIMULATION TESTBED FOR THE DEVELOPMENT OF ATM/C AUTOMATION SOFTWARE

3.1 Overview

In the course of this work we were forced to deal again and again with "human factors" aspects of the problem of ATM/C automation in spite of the fact that the original intention was to study and develop automatic scheduling and flight plan generation algorithms which would be tested using a fast time simulation. It was concluded finally that in all cases algorithmic and software development has to take into account the method of interaction with the ATM/C controller. It was therefore apparent early in the research effort that, in order to effectively test and demonstrate any of the automation functions described in the previous chapter, a real time, interactive terminal area simulation facility was necessary. A terminal area simulation facility, TASIM, was designed and implemented to fulfill this purpose. It is a substantial extension of the original work done by Heinz, [HEI 76].

TASIM captures in great detail the performance of all elements of a terminal area ATM/C system. Particular emphasis was placed on the interactions between various elements and on the flow of data within the ATM/C system. To insure correct and easily identifiable program logic,

a top-down approach to software development was adopted. Composite design techniques were used to produce a system consisting of highly independent functional parts (software modules) interconnected through clearly defined interfaces and performing clearly defined functions. The final product is a realistic and flexible simulation facility which can be responsive to the needs of human factors research. It therefore provides a vital research tool for the development of ATM/C automated decision support.

This chapter will give a brief description of TASIM and highlight the features that are of consequence in realistically reproducing the environment within which ATM/C automation functions have to perform.

3.2 The Hardware Environment

The current hardware diagram for TASIM is shown in figure 3-1. A VAX-11/780 provides the main computing power and houses about 95% of the software. A PDP-11/34 computer is used to generate the traffic (radar) display and communicates via data link with the VAX-11/780 only to receive information on active aircraft in the simulation or on controller inputs requesting some type of display restructuring function to be performed. In addition the facility currently includes an ATM/C controller station and a simulation monitoring station.

The ATM/C controller station uses a MEGATEK calligraphic display computer as the TD and a VT-100 video terminal as the CTID. It is the

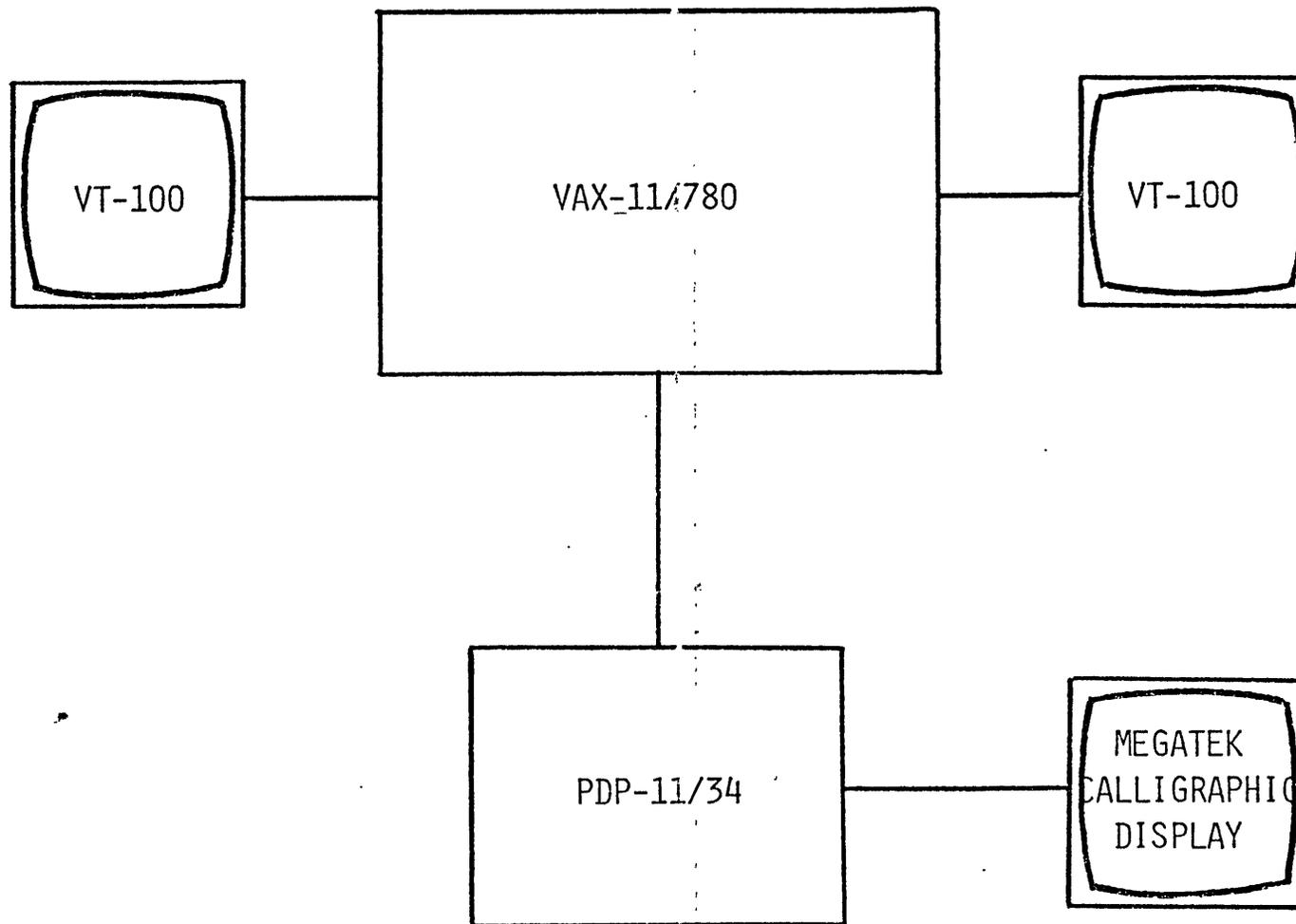


Figure 3-1. Current TASIM Hardware Diagram.

primary man-machine interface. The traffic movements are displayed on TD in ARTS III format, while the CTID (Controller Traffic Information Display) is used to display supplementary tabular information generated by various automation functions. The VT-100 keyboard is the data entry device allowing real-time interaction between the ATM/C controller and the simulation software functions.

The simulation monitoring station, consisting of a second VT-100 video terminal, is currently used only to initialize the data base as well as start and end the simulation run. Since it also allows real-time interaction with the simulation software, its capabilities can be expanded in the future to include modification of the data base in real time. This capability will allow direct control over traffic levels, weather and wind conditions, and can be used to bring about unusual situations that merit close investigation (e.g. procedures for changing runways due to shifting wind direction, etc.).

In the near future TASIM will be moved to a VAX-11/750 with three SANDERS Graphics-7 displays as TD's and three Texas Instruments 940 terminals as CTID's, [PAR 82].

3.3 Major System Components and Interfaces

The simulation software consists of 4 major components, called processes, as shown in figure 3-2. Each process is a separate "stand-alone" computer program operating in parallel and independent of

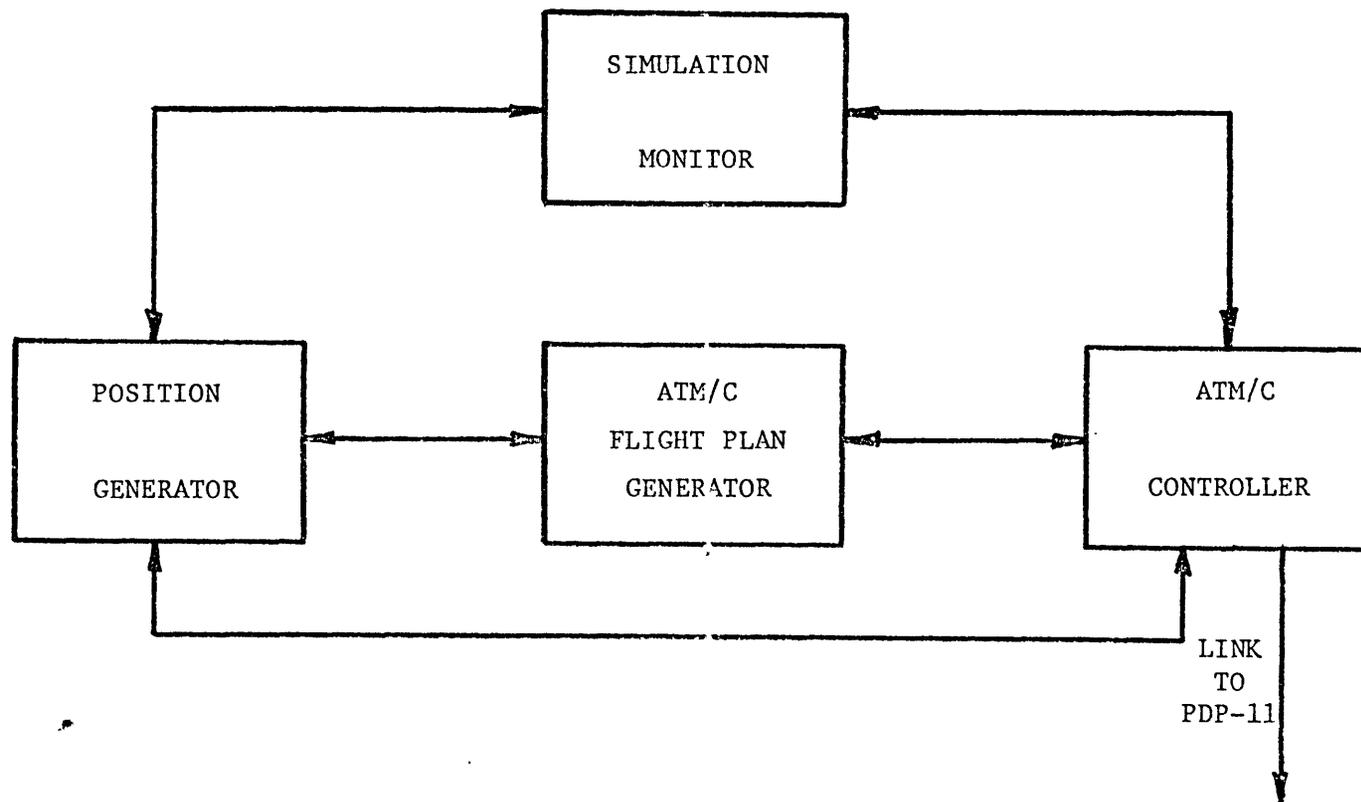


Figure 3-2. TASIM Software Processes.

the others. With the exception of the traffic display program which operates on the PDP-11/34 computer, they all share the VAX-11/780 CPU. Within the VAX computer each process communicates with the others via virtual I/O devices called mailboxes. Each process has a mailbox associated with it. When a message is received in the mailbox, a specific function is performed depending on the message contents and possibly the process which sent the message. A common, memory-resident data base is shared by all VAX-11 processes.

The Simulation Monitor Process oversees the simulation execution. Prior to the beginning of real-time operation, it initializes the global data, initializes and insures the proper loading of subordinate processes, creates the communication mailboxes, and accepts operator input to start the real-time execution. During the execution it is responsible for the timing and allocation of CPU time to various processes as required. In addition it collects, and optionally displays, statistics on CPU usage, and the status of all other simulation processes.

The Position Generator Process is responsible for maintaining and updating the state (e.g. position, speed, altitude, etc.) of all active aircraft in the simulation. It provides realistic dynamic models of the Aircraft Control Systems and the Air-to-Ground Data Acquisition System (surveillance radar and tracking processing) depicted in the overall schematic diagram of the terminal area ATM/C system (figure 2-1).

The Air Traffic Controller Process functions as the primary interface between the simulation software and the ATM/C controller. It accepts and processes ATM/C controller entries, and displays ATM/C information such as the current sequence of runway operations, computer generated commands for aircraft to follow, and simulation operating statistics (e.g. average aircraft delay, etc.). In addition, it is responsible for maintaining and updating the traffic situation on the Traffic Display in cooperation with the PDP-11 software. In the current implementation the Command Activator also resides in this process.

The ATM/C Flight Plan Generator implements three major functions: Nominal Flight Plan Generation, Runway Scheduling, and Traffic Flight Plan Generation. It determines the optimal schedule of runway operations and the recommended flight plans to implement that schedule. The flight plans are then translated into ATC commands by the Command Generator which resides in this process. The commands are subsequently dispatched by the Command Activator to the ATM/C controller for validation and subsequent transmission to the aircraft.

The choice of functions performed by each process closely matches the real ATM/C environment where several systems operate in parallel and communicate through various links. Of course, a computer is basically a sequential machine and to accomplish true parallel processing would require several communicating computers. Fortunately by allocating small intervals of CPU time to each process, it is possible to simulate parallel processing at least macroscopically. Furthermore, the details

of how CPU time is allocated to each process are the responsibility of the VAX operating system while the Simulation Monitor can provide the framework for CPU allocation by adjusting process priorities as necessary.

3.4 The Simulation Monitor Process

The Simulation Monitor process controls and monitors the overall simulation execution. It does not perform any Air Traffic Control function, but creates the real-time environment necessary for the remaining processes to operate properly. It is also responsible for initializing operations. The overall flow chart of the simulation monitor is shown in figure 3-3.

3.4.1 Data Base Initialization

We distinguish between two types of data in the simulation: global data and local data. Global data are shared (i.e. can be referenced and/or modified) by all processes in the simulation. Local data can only be accessed by a single process. The major part of TASIM data base is global. Local data generally consist of working areas and other such data.

Global data are categorized into 4 groups:

1. Process data
2. Dynamic data

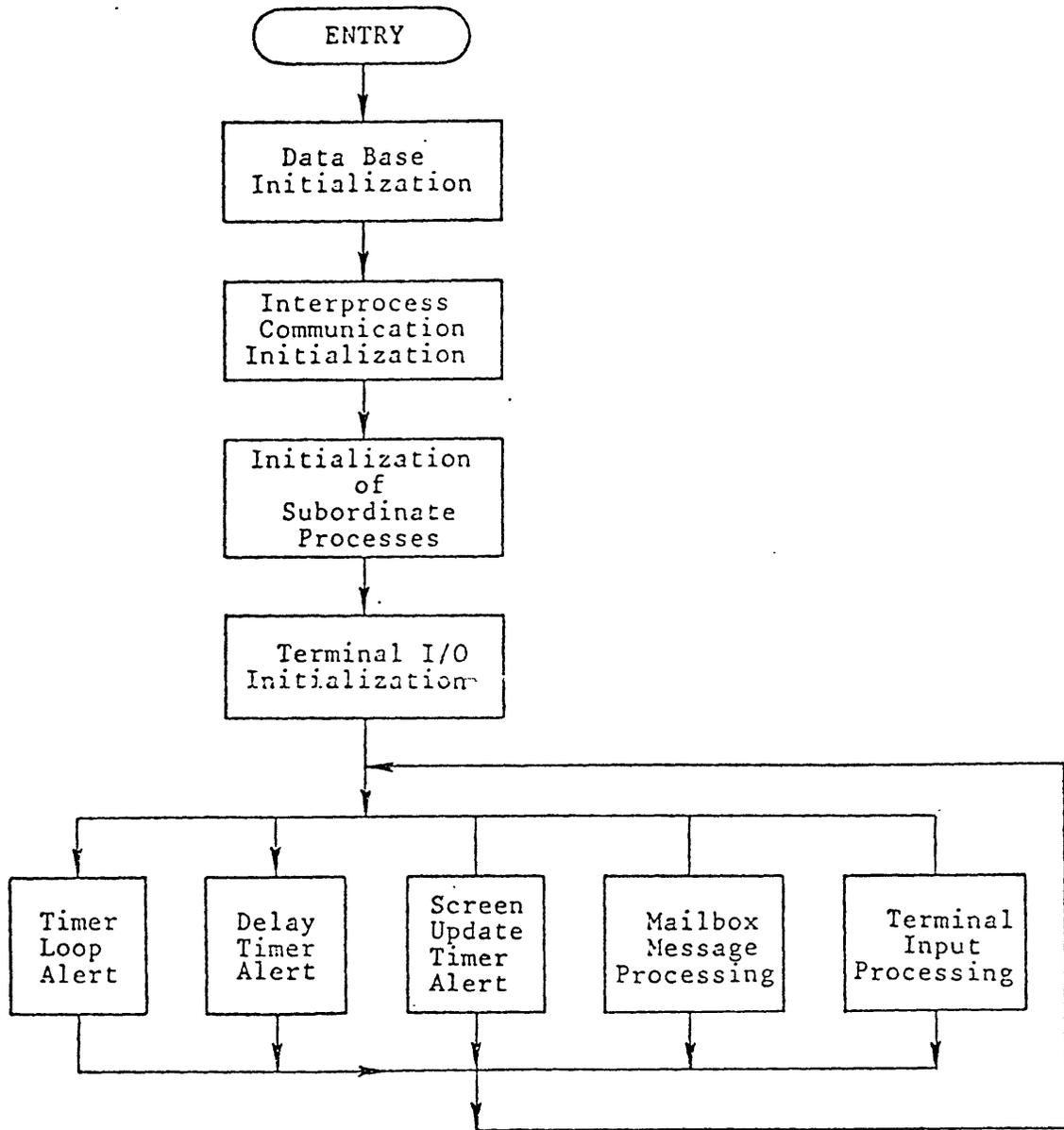


Figure 3-3. Top Level Flow-Chart of the Simulation Monitor Process.

3. Static data

4. Screen data

Process data contain information required by the Simulation Monitor to control and monitor the various processes, as well as data required by each process to control its operation. The former include process identification number, process name, execution priorities, CPU utilization statistics, etc. The latter include process specific information such as current simulation time (for the Simulation Monitor process), number of currently active aircraft and total number of aircraft (for the Position Generator process), etc.

Dynamic data consist of aircraft "files" and radar "files". Each aircraft file contains pertinent information on one active aircraft in the simulation. Four types of information are maintained:

1. Aircraft ID data, which include the aircraft type, flight number, transponder code, and other such data typically found in a flight strip.
2. Aircraft state data, which include position, altitude, speed, heading bank angle, and other such information describing the the instantaneous dynamic state of the aircraft. Actual values as well as ^t values indicated by the various airborne or ground instruments are maintained.

3. Aircraft command data, which include commanded altitude, speed, heading and other such data describing clearances the aircraft has received or is due to receive in the future.
4. Aircraft scheduling data, which include current sequence number, the assigned runway, the scheduled time of arrival at the runway, as well as other such data describing the optimal runway schedule under the current objective function.

Each radar file contains surveillance information provided by one surveillance radar in the simulated airspace. Any number of radars can be simulated though typically only one is used as the primary source of position information at any given time.

Static data consist of reference information which describes the simulation environment and remains unchanged throughout the run. This group includes data describing the simulated airspace (i.e. location of airports, runway description, location and type of navigational aids and radars, nominal departure and approach routes, etc.), the characteristics of simulated aircraft types, the traffic levels and mix, etc. Note that even though the traffic levels and the traffic mix may change with time, they are considered static data since they are determined apriori.

Screen data consist of information specially formatted for display on the CTID screen. This group contains information already found elsewhere in the simulation data base. Here, however, this information is in a format suitable for quick display. Even though this procedure requires additional computer memory, it was adopted since it conserves CPU time which would have been required to repeatedly reformat the same information whenever needed for display, and in a real-time environment trading-off memory for CPU time is advantageous.

3.4.2 Interprocess Communication

Interprocess communication is the "central nervous system" of the simulation and consequently, design of the communication protocol is very important for proper operation as well as for maintaining a flexible and expandable system.

Interprocess communication is accomplished via virtual I/O devices called mailboxes. A process (sender) leaves a message in another process' mailbox (receiver) providing vital information and (usually) requesting for some function to be performed. The receiver reads its mail, performs the requested function(s) and (optionally) advises the sender of the outcome and/or completion of the operation. The message minimally contains the message code and the identification of the sender process. It may contain additional information regarding the request when such information is not readily accessible from the global data base. For example, when some function has to be performed on a specific

aircraft, the aircraft identification is included in the message, whereas additional information that may be required can be retrieved from the global data base by the receiving process.

This method of interprocess communications allow both asynchronous and sequential processing of messages to be easily implementable. This is very important in simulating the ATM/C environment where some functions require immediate processing while others should be deferred either because some higher priority function takes precedence or because certain conditions have to be met before the function can be successfully processed. A good example of asynchronous processing is the message sent by the Simulation Monitor to the Air Traffic Controller process notifying it of the availability of data to be sent to the Traffic Display Driver (i.e. to the PDP-11 computer). To insure smooth operation and prompt display of the latest information on the positions of aircraft in the system, this function has to take precedence over for example, the processing of input that the ATM/C controller may have entered on the keyboard. The latter function is resumed when the TD update has been completed. In contrast, when the ATM/C controller has manually entered or has validated a command to some aircraft, the Position Generator process receives a message in order to enter the new information in the appropriate aircraft's command data. The processing of the request is waived by the Position Generator until the end of the aircraft state update cycle since the latter is the highest priority function of the two.

3.4.3 Subordinate Processes and Execution Control

The simulation monitor is responsible for loading into memory and preparing for execution all the subordinate processes which comprise TASIM. Currently these are the Position Generator, the ATM/C Controller process, and the Flight Plan Generator. The information contained in the Process data (described in the previous section) govern the loading of all processes.

The functions performed by the simulation can be divided into two broad categories, periodic and aperiodic. Periodic functions are triggered by the internal clock and are performed at specific time intervals. All periodic functions are controlled by the Simulation Monitor which has sole responsibility for timing the software and achieving real-time performance.

The logic according to which the Simulation Monitor controls the execution of periodic functions is uniform and independent of the specific function performed. This is achieved by assigning processes into "timer chains" which define the sequence in which each process will be activated in response to a specific timer alert. Within each process there is an explicit sequence of functions (or possibly a single function) which is performed in response to the timer alert message sent by the Simulation Monitor. Upon completion of the requested function(s) each process sends a termination message to the Simulation Monitor which causes the latter to look for and activate the next process in the timer

chain. When the last process has been activated, the Simulation Monitor waits for the expiration of the time interval at which time the timer chain is restarted. A process may be part of more than one timer chain and perform different functions depending on which timer chain caused the process activation.

Aperiodic functions are triggered by some simulation event which is not generally guaranteed to occur at any specific time or with any regularity. The performance of such functions does not necessarily involve the Simulation Monitor but is triggered by a direct message by the process which detected the event to the process which is responsible for performing the function in question. The update of the aircraft command data triggered by a manually entered command, is a good example of an aperiodic function performed by the Position Generator process due to an event detected by the Air Traffic Controller process.

The execution logic is thus layered into several functional levels with each level controlling the execution of the level immediately below and controlled by the level immediately above it. Within each level the functions performed require minimal awareness of the chain of events that triggered their activation. This allows easy modification of function at any level by changing the sequence of subfunctions activated and/or inserting new functions in that chain. Furthermore it allows functions to be included in more than one level in this structure and perform as parts of more than one higher level functions. In the

current version the simulation has a single periodic timer chain.

Currently the periodic functions of TASIM are:

1. Traffic Generation
2. Aircraft Dynamics update
3. Air Data system update
4. Aircraft Navigation system update
5. Surveillance system update
6. Aircraft Status update
7. Command Activator
8. Traffic Display data update

The first six functions are performed by the Position Generator process while the last two are performed by the Air Traffic Controller process. In the current configuration these functions are performed every 4 seconds which corresponds to the typical period of revolution of the radar antenna in the terminal area. This need not be the case however since a time chain can be executed at any period as long as there is enough CPU time for all the required computations to be performed.

Currently the aperiodic functions of TASIM are:

1. Runway Scheduling
2. Keyboard Input Editing and Processing
3. CTID update

Runway scheduling is performed by the Flight Plan Generator process, while the other two are performed by the ATM/C controller process.

3.4.4 Initialization of Terminal I/O

Terminal initialization prepares the terminal for real-time input and output operation. It includes formatting of prompts as well as standard computer responses, setting the video screen in the proper operating mode for display of tabular information and initializing the internal data required to control the cursor position and movement. Since the general purpose terminal driver provided by the VAX-11/780 operating system was not adequate to perform the special functions required by the simulation (e.g. direct cursor control, item selection, editing of input, etc.), a special purpose driver was designed and implemented. The terminal driver was designed specifically to suit the needs of the interface between the ATM/C controller and the simulation and will be discussed in connection with the ATM/C Controller process (section 3.6.6).

When the initialization phase is completed, the Simulation Monitor process waits for input from the operator requesting the beginning of real-time operation. Upon entering the real-time mode, the program initiates the Position Generator-ATM/C Controller timer loop. Subsequently the wait state is entered again. At this point however a number of occurrences require the Simulation Monitor's attention and

will cause it to exit the wait state. These are described in the following sections.

3.4.5 Timer Loop Alert

The timer loop alert indicates that it is time to start a new cycle of the periodic timer loop. Figure 3-4 shows the processing of timer loop alerts. Even though only a single timer loop is defined currently the logic has been designed to handle any number of timer loops.

The loop is first identified and if the last process in the chain has completed execution the procedure for initiating a new timer loop takes place. This includes: (i) requesting (from the operating system) the delivery of a new alert at the end of the time interval appropriate for the timer loop in question, (ii) sending a START message to the first process in the timer loop, (iii) collecting CPU usage and other statistics from this timer loop, and (iv) reentering the wait state. If the last process in the timer loop has not completed processing, the initiation of the new cycle must be delayed. Such situations should not occur during normal operations since they indicate that real time operation cannot be achieved due to CPU unavailability. The required delay in the initialization of a new cycle is implemented by: (i) requesting a delay timer alert at the end of a much shorter time interval (currently set at 1/100 of a second), (ii) initializing the compilation of delay statistics, and (iii) re-entering the wait state.

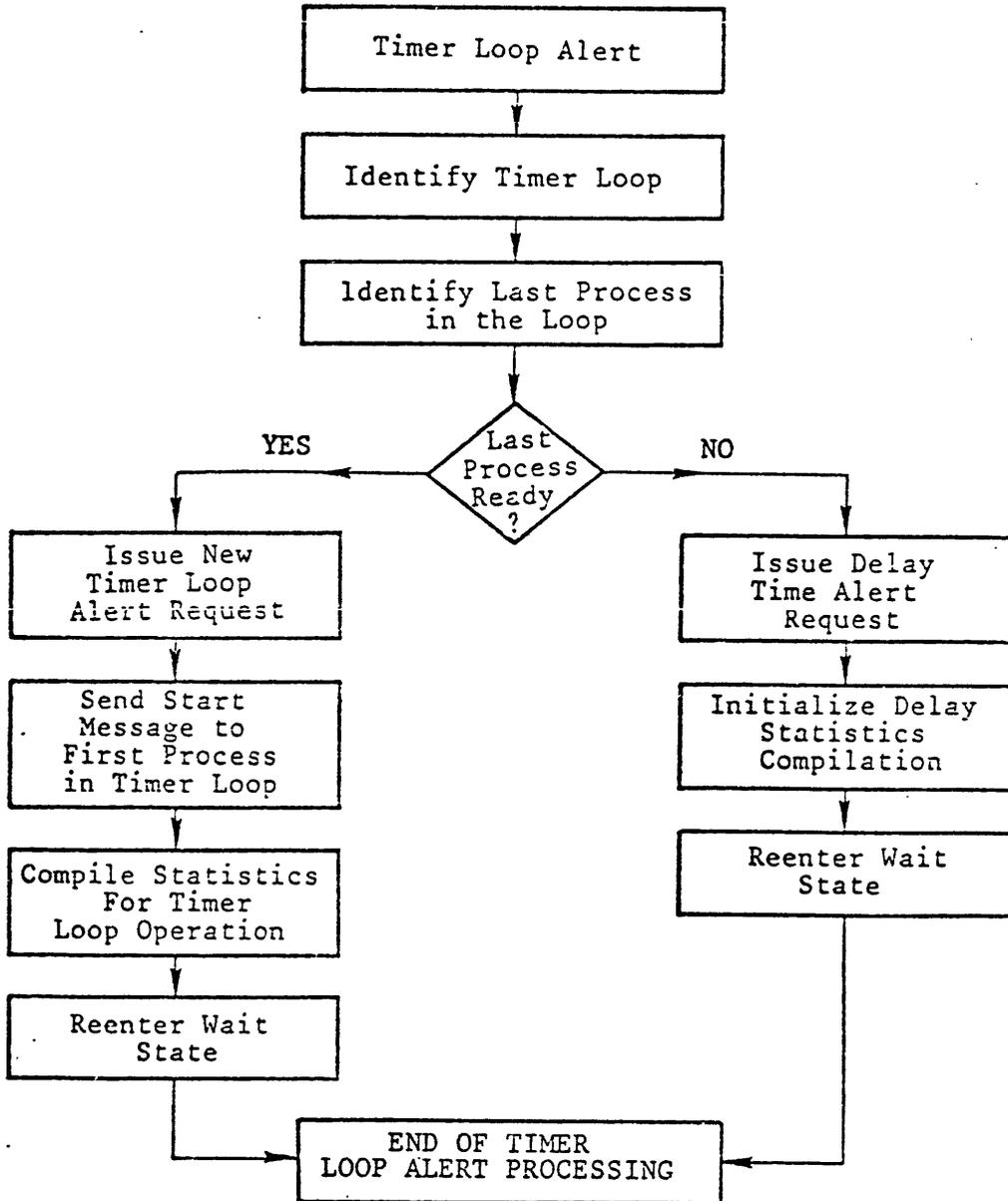


Figure 3-4. Timer Loop Alert Processing Detail.

3.4.6 Delay Timer Alert

The delay timer alert indicates that the initialization of a timer loop has been delayed due to CPU unavailability. This situation is of course undesirable and is not expected to occur during normal operation. Indeed, even in the time sharing environment in which the system operates currently, the simulation has not incurred delays unless there has been a high number of other users (e.g. 15-20) on the system. This indicates that the current CPU usage is well below the VAX-11/780 capabilities.

The processing logic for the delay timer alert is depicted in figure 3-5. The timer loop that caused the alert is identified, statistics are updated to account for the additional delay interval, and if the last process in the loop has completed execution, the new cycle for this timer loop is initiated. Otherwise the timer loop initialization is further delayed.

3.4.7 Screen Update Timer Alert

This alert occurs periodically and causes the update of information on the VT-100 video screen if the displayed information has been superseded. This alert is common to the Simulation Monitor and the ATM/C Controller process. Since it is an integral part of the display interface, its discussion will be presented in connection with the ATM/C Controller process (section 3.6.6).

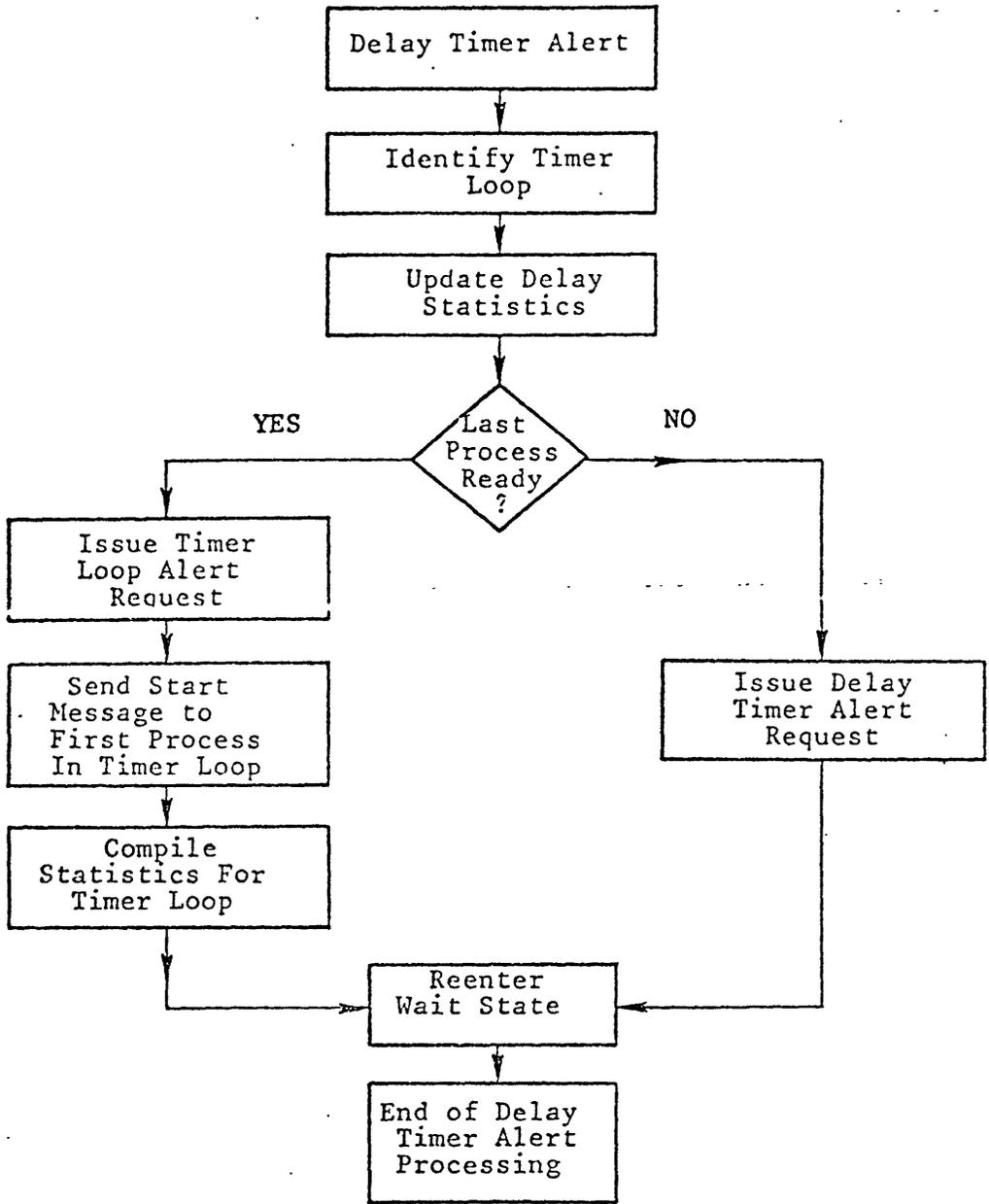


Figure 3-5. Delay Timer Alert Processing Detail.

3.4.8 Mailbox Message Processor

When a message is placed in its mailbox the Simulation Monitor exits the wait state in order to process and respond to the message. The processing depends on the type of message received and on the sending process. The following message types are currently implemented:

1. READY message

This message can be sent by any process in a periodic timer loop to inform the Simulation Monitor that processing requested via the START message (see sections 3.5.1 and 3.6.1) has been completed and thus the next process in the timer loop, if any, can be STARTed. If the sending process is the last one in the timer loop, no processing is required since the timer loop cycle will be initialized when a timer alert or a delay timer alert occurs. If the process is not the last one in the loop, a START message is sent to the next process. The wait state is reentered in either case.

2. EXIT message

This message is received by the Simulation Monitor when the sending process is about to end execution. While in the normal real-time mode, reception of the EXIT message indicates that the process has encountered some error

condition and thus an abnormal termination of the run is imminent. If the system is already in the termination mode, this message is simply a reply to an EXIT message sent by the Simulation Monitor to all processes as part of the rundown procedures. In the normal mode, the Simulation Monitor outputs the identification of the process that sent the message as well as the error condition code. It then enters the termination mode. EXIT messages are sent to all processes that are still on-line and the wait state is entered. If an EXIT message is received in the termination mode, the program checks if all processes have exited. If not, it waits for a new EXIT message. When all processes have exited, the Simulation Monitor stores the global data base, deletes all mailboxes, deassigns all I/O channels and exits. The processing is shown schematically in figure 3-6.

3.4.9 Terminal Input Processing

When input from the terminal is available, the process exits the wait state to parse the input and perform the desired function. Two commands are currently implemented:

1. START

This command initiates the real-time operation of the simulation. Optionally, the simulation time at which the

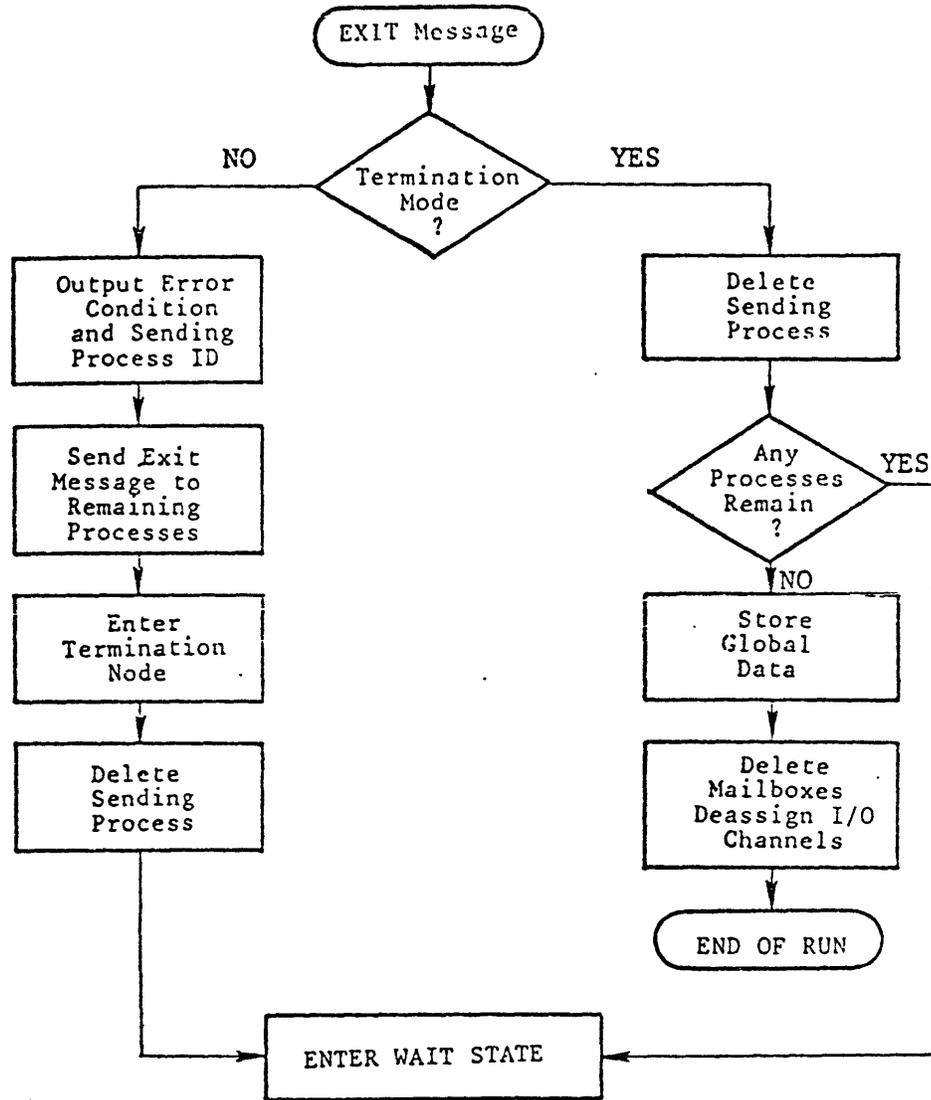


Figure 3-6. EXIT Message Processing Detail.

real-time operation should begin can be specified. In that case the simulation operates in the FREERUN mode until the specified time is reached. At that point the real-time operation is initiated. This mode permits TASIM to quickly pass through the "transient state" of traffic building up if the steady state performance is of interest.

2. END

This command causes the process to enter the termination mode, send EXIT messages to all other processes, and wait for replies before exiting (see previous section).

3.5 The Position Generator Process

The Position Generator process simulates the movements of the aircraft targets in the simulated airspace as well as the surveillance radars and the navigation equipment. It consists of seven major subsystems:

1. Mailbox Message Processor
2. Traffic Generator
3. Aircraft Status Update
4. Aircraft Dynamics model
5. Air Data System model

6. Navigation System model
7. Surveillance System model

The surveillance system drives all the other components since all other functions are performed once every revolution of the radar antenna (see discussion of the Simulation Monitor process in the previous section).

3.5.1 Mailbox Message Processor

The operation of the Position Generator process is controlled by messages received in the mailbox. The mailbox message processor is responsible for interpreting those messages and invoking the required functions. The messages defined in the current configuration are:

1. START

This message is sent by the Simulation Monitor and causes a new update cycle to be started. All subsystems are invoked sequentially. When the update is complete, a ready message is sent to the Simulation Monitor's mailbox.

2. READY

This message is sent by the Flight Plan Generator process to inform the Position Generator that processing (i.e. runway scheduling in the current configuration and traffic plan generation in the future) has been completed. It

allows the Position Generator to coordinate its requests for rescheduling (see next section).

3. EXIT

This message is sent by the Simulation Monitor to signal the end of the run. It causes the position generator to pause execution and to respond with an EXIT message before exiting.

3.5.2 Traffic Generation Model

The traffic generation model is designed to realistically duplicate in detail the processes which generate traffic in the terminal area. Traffic is generated due to demand for landing at, or taking off from airports within the terminal area under consideration. In the simulation each airport is characterized by demand rates as well as traffic mixes. These are separate for landings and for takeoffs. The generation of landings and of takeoffs are assumed to be independent Poisson random processes. The time of generation is assumed to be the earliest time the ATM/C controller has knowledge of some aircraft's intentions to land at or takeoff from the airport

For each airport, an aircraft (takeoff or landing) is generated during each simulation update interval Δt with probability

$$p_i = \lambda_i \Delta t$$

where λ_i is the demand rate for this type of operation at an airport i (number of aircraft per unit time)

It is well known that the resulting generation times will form a Poisson process provided that Δt is small enough compared to the inverse of the demand rate, $1/\lambda_i$ [DRA 67]. When this process results in the generation of an aircraft, the current simulation time is assigned as the aircraft's generation time.

The aircraft type is determined based on the aircraft mix applicable to each airport. The mix for landing aircraft may be different for that of takeoffs. Variation in the mix over time may be modelled as a step function or as a piecewise linear function.

Fixes are described in the simulation by their latitude and longitude, the types of aircraft that are can be assigned to the fix, the relative frequency with which aircraft taking off or landing at an airport use the fix, etc. In addition each fix has several altitudes that can be assigned to aircraft using it.

In order to determine the fix for the newly generated aircraft, the model first determines all eligible fixes. In order to be eligible, the fix must be active and able to be assigned to the type of aircraft which has just been generated. Once the list of eligible fixes has been compiled, the relative frequencies with which each is used by aircraft operating at the airport in question are normalized. The

assigned fix is then randomly selected from the resulting probability distribution.

In some cases the interval between the generation of two successive landings bound for the same entry fix will be such that the ATC requirement for horizontal separations will be violated. In such cases the traffic generation model simulates the actions that might have been taken by the enroute controller in order to provide adequate separations. Let n be the number of available altitude levels at some specific entry fix. The model assumes that lowest altitude is the most desirable and the highest is the least desirable. For each altitude i , the time t_i of the last aircraft passage is stored. At any time t the altitude is called open if

$$t_i + t_{\min} \leq t$$

where t_{\min} is the minimum time separation between two co-altitudinal aircraft.

The lowest open altitude at the time the aircraft is expected to arrive at the entry fix is assigned. If no open altitude exists at that time, the time of arrival at the fix is revised to coincide with the earliest time an altitude is open. The inherent assumption in this model then, is that the enroute controller is aware of the situation at the entry fix and delays incoming traffic in order to insure that aircraft are handed-off to the terminal area controller with proper separations.

If at least one aircraft has been generated during a cycle, the Flight Plan Generator process is alerted through a mailbox message to incorporate the new aircraft in the automated decision support functions. Currently this message triggers the Runway Scheduling function which will determine the new runway schedule.

3.5.3 Aircraft Status Update and Handoff Processing

This function is responsible for maintaining and updating the status of all aircraft as they pass through various phases of their "life" in the simulation. At any time the status of any aircraft is

1. Dormant
2. Active
3. Inactive
4. Marked for deletion

Upon generation all aircraft are dormant. Dormant aircraft are ignored by all functions of the simulation with the exception of the aircraft status update function and the Runway Scheduler. When the activation time is reached for some aircraft, it is activated and becomes a full participant in the simulation. Aircraft remain active until they reach their final destination at which time they are deactivated. Landings become inactive shortly before touchdown on their assigned runway while departing traffic is deactivated upon reaching their assigned terminal area exit fix. Upon deactivation of an

aircraft, various simulation subsystems have to be given a chance to purge their (local) data bases discarding the now useless information. When this is completed, the aircraft is marked for deletion at which time statistics are collected, the aircraft global data are saved if needed, and the aircraft is finally deleted from the system.

This function also handles automatic handoff initialization and activation. The position and direction of an aircraft determines whether it is appropriate to start a handoff procedure. The ATM/C controllers that are involved are identified and alerted through ATM/C messages. If the target ATM/C controller has disabled the automatic handoff acceptance mode, the handoff status for the aircraft is maintained until the handoff is explicitly accepted. Reminder messages are sent if a prespecified time interval is elapsed without reception of handoff acceptance.

3.5.4 Aircraft Dynamics

This function is responsible for maintaining and updating the dynamic state of all aircraft in the simulation, moving them through the simulated region in response to internal and external forces. Internal forces are generated to bring the measured aircraft state into alignment with the commanded state and include power changes and control surface variations. External forces include the effects of wind direction and magnitude. The aircraft model used accurately describes the functional relationships between various aircraft components, and includes time

lags present in the longitudinal, lateral and vertical response of each aircraft. The model was implemented by Heinz and is described in detail in [HEI 76] and [HOF 72].

3.5.5 Air Data System Update

This function manages and updates readings of the available airborne instruments such as the compass/gyro (heading indicator), the airspeed/mach number indicators, the altimeter, and vertical speed indicator. The measurement errors for all the instruments are based on models developed by Hoffman [HOF 72]. The model was implemented by Heinz [HEI 76].

3.5.6 Aircraft Navigation System

This function is responsible for maintaining current readings of the airborne navigational equipment for each simulated aircraft and simulating the navigation system errors. By and large, aircraft navigation today is performed with the use of radio navigational aids. Two different devices are modelled: (i) Very High Frequency Omnidirectional Range (VOR), which indicates the magnetic bearing from the station to the aircraft, and (ii) Distance Measuring equipment (DME-TACAN), which indicates the slant range between the station and the aircraft. As is generally the case, the simulation assumes that VOR and DME-TACAN transmitters are collocated in VOR/DME or VORTAC installations so that both the range and the bearing from a single known geographic

location is available. Each aircraft is capable of independently tracking two VOR/DME or VORTAC stations. This allows a variety of navigation modes to be simulated including area navigation and VOR radial interception while tracking a different VOR course. The models used for aircraft navigation systems and their errors were developed by Hoffman [HOF 72]. The model was implemented by Heinz [HEI 76].

3.5.7 Surveillance System

This function models the radars that provide Ground Control with aircraft position information. Two types of surveillance systems are generally employed in tracking airborne aircraft. Surveillance radars utilize reflected energy from the ground transmitter (often referred to as skin tracking). Beacon trackers interrogate a transponder on the aircraft which transmits a coded reply, consisting of the aircraft ID and sometimes altitude information. Normally the two systems are operated in parallel, with the surveillance radar serving as a backup to the beacon system. The simulated radar is described by a set of parameters which determine its performance. Table 3-1 gives the values for those parameters that are used to describe the terminal area and enroute versions of the current civil beacon system known as ATCRBS (Air Traffic Control Radar Beacon System). Note that tracking algorithms have not yet been implemented in TASIM.

TABLE 3-1

ATC Radar Beacon System Parameters

<u>Parameters</u>	<u>Terminal</u>	<u>Enroute</u>
Max range	60 nmiles	200 nmiles
Elevation coverage	0 to 45 degrees	0 to 45 degrees
Scan rate	15 RPM	6 RPM
Range bin width	1/16 nmiles	1/4 nmiles
Azimuth bin width	0.088 degrees	0.225 degrees

3.6 The ATM/C Controller Process

The ATM/C Controller process is responsible for accepting and processing real-time commands input by the ATM/C controller, and for managing the controller tabular information display (CTID) and the traffic display (TD). The traffic display software reside partly in the VAX-11/780 and partly in the PDP-11/34. It will be described here, however, since it is an integral part of the ATM/C Controller process. In the current configuration the command activator is a function of the ATM/C Controller process.

Figure 3-7 presents a top level functional diagram of the ATM/C Controller process and identifies the major software components. The hardware (VT-100 video terminal and MEGATEK calligraphic display computer) are also depicted in the figure.

The mailbox message processor is invoked by the reception of a message in the mailbox. In turn, it invokes the Traffic Display Data Management software and the Command Activator when a START message is received. The Traffic Display Data Management software formats and sends the appropriate aircraft data to the Traffic Display driver (resident in the PDP-11/34) which updates the positions of the aircraft on the MEGATEK screen. The Traffic Display driver also responds to data sent by the Input Processor to modify the screen characteristics (displayed range, displayed altitudes, etc.). The CTID Display software is triggered periodically by a timer alert and updates the information

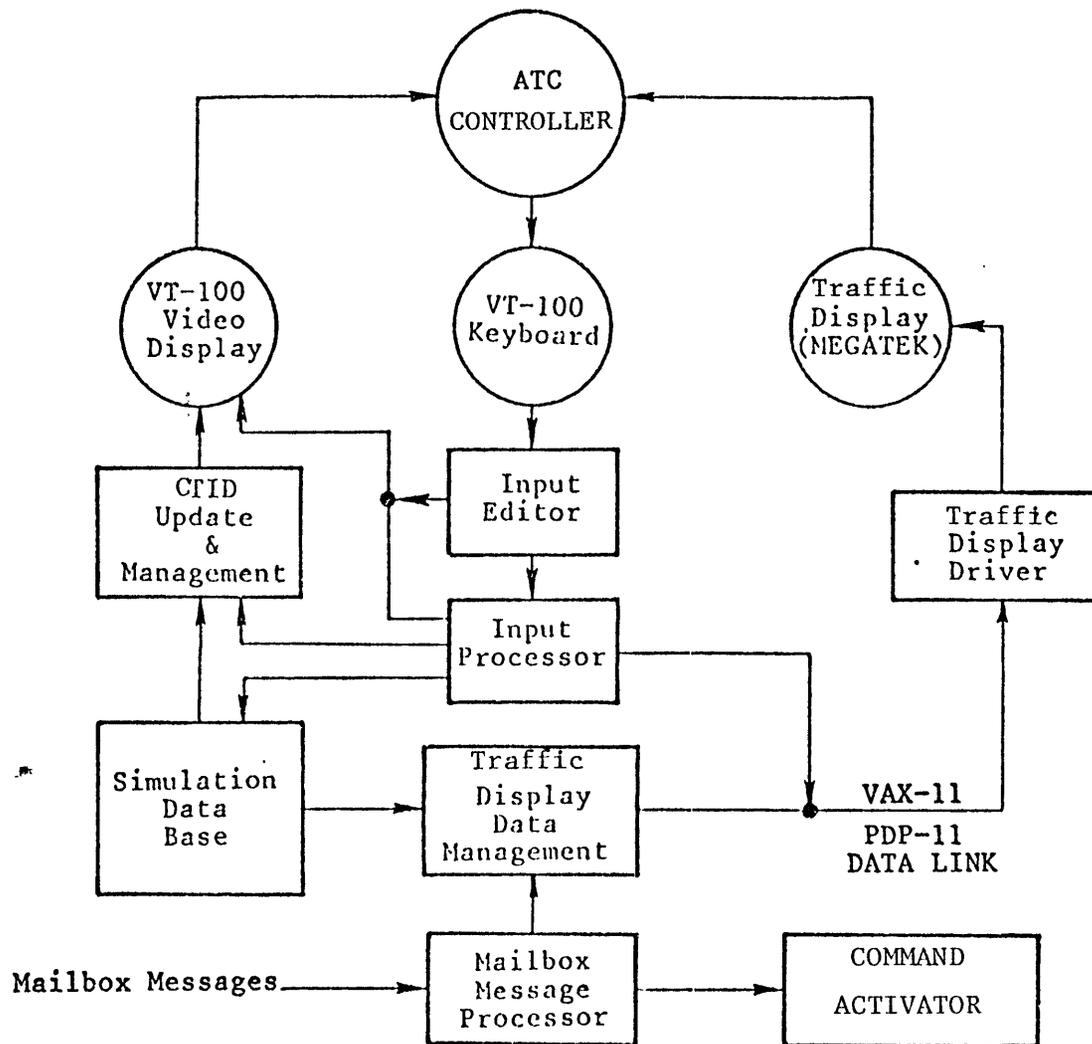


Figure 3-7. Functional Diagram of the Air Traffic Controller Process.

on the VT-100 video display screen. The Input Editor is triggered by controller inputs. Each input character is processed separately as it is entered. This allows various editing functions to be performed on the input line by defining special function keys. When a line ending character is entered, the input editor transfers the command line to the Input Processor which is responsible for identifying and performing the requested function. The Input Processor displays warning and error messages on the VT-100 screen to inform the ATM/C controller in case of incorrect, ambiguous or unrecognized input.

3.6.1 Mailbox Message Processor

Two messages are currently recognized by the Mailbox Message Processor:

1. START

This message, sent by the Simulation Monitor, directs the ATM/C controller process to initiate an update of the aircraft positions on the radar display and to invoke the Command Activator function. When both these functions are complete the process responds with a READY message sent to the Simulation Monitor. This currently ends the only timer loop of the simulation (see section 3.4.5).

2. EXIT

This message, sent by the Simulation Monitor, signals the

end of the run. The ATM/C controller process deassigns the I/O channels, associates with the VT-100 and the mailboxes, stops the display driver software on the PDP-11/34 computer, responds with an EXIT message sent to the simulation monitor, and exits.

3.6.2 Command Activation

This function is responsible for maintenance and timely dispatching of commands to the aircraft. Piloting commands are generated either by the Command Generator based on the aircraft flight plan or are directly input by the ATM/C controller. For uniformity the commands generated automatically are in the form of alphanumeric strings and in the same format as the ones entered by the ATM/C controller. Processing of both types of commands is therefore identical.

Two concepts for command management can be identified. In present ATM/C systems there is no precise future planning so that the need for issuing a piloting command at some future point in time cannot be fully anticipated. Even when, for instance, the ATM/C controller knows that soon some aircraft will be required to make a turn to intercept the ILS, he does not have the means to determine the exact time or location at which this command should be executed. Furthermore it is not at all clear that the pilot will accept a command to turn to some heading, say, 90 seconds from now when he is busy preparing for landing. As a consequence the ATM/C controller waits and issues the command when the

aircraft reaches a certain point in space and the pilot has to execute it immediately. We refer to this process as "queueing the commands on the ground".

We can, by contrast, visualize the opposite process, namely "queueing commands in the air". Here commands are transmitted to the aircraft in advance and it is the pilot's responsibility to execute them at the proper time. We distinguish between time-triggered and location-triggered commands, depending on whether the command execution starts at a specific point in time or at a specific point in space. An IFR flight plan is an example of queueing location-triggered commands in the air since it is the responsibility of the pilot to follow the approved route. A missed approach procedure is an example of this process in the terminal area environment. The only time-triggered command sequence in today's ATM/C system is the execution of a holding pattern where the pilot is responsible to perform one minute turns followed by 1,2 or 3 minute straight legs. In advanced ATM/C systems incorporating four dimensional flight planning, time-triggered command queueing in the air is certainly feasible (from a technological point of view) when coupled with a digital air-to-ground data link and airborne time-triggered autopilots.

The simulation was designed with the capability to handle both time-triggered and location-triggered commands. This is accomplished by associating four different time values with each command:

1. the scheduled time of transmission
2. the actual time of transmission
3. the scheduled time of execution
4. the actual time of execution

The interpretation of each of these depends on the method employed by the ATM/C system. When commands are queued on the ground the difference between the scheduled and the actual time of transmission is caused by ATM/C controller workload and/or congestion of the communication links. By contrast, the difference between the actual time of transmission and the actual time of execution is the pilot response time and is therefore associated with pilot workload. When commands are queued in the air, the difference in transmission times (actual and scheduled) is zero as long as the command is transmitted before it is scheduled to be executed. The discrepancy in execution times in that case measures the capability of the pilot and/or aircraft autopilot to execute the commands promptly.

There are distinct disadvantages in queueing commands in the air even though this is feasible for advanced automated ATM/C systems. The most compelling of all is the need to develop systems that are compatible (to the greatest extent possible) with existing equipment. Queueing commands on the ground only requires advanced equipment on the ground (namely the computers and computer programs that implement the advanced ATM/C functions) and is capable of operating with absolutely no advanced capability requirements onboard the aircraft. Second, by

transmitting commands to the aircraft well in advance we either, (i) deprive the system the capability to revise commands that have not yet been executed if the changing traffic situation warrants it, or (ii) we create congestion on the communications link by transmitting and subsequently revising or cancelling commands.

The ATM/C system concept described here queues commands on the ground thus avoiding all these problems. Aircraft flight plans, and therefore commands, are subject to change until some short time interval before they are scheduled to start execution. At that time a message is sent to the ATM/C controller to inform him that the command will be transmitted to the aircraft shortly and to request validation if he has not done so yet. This interval is chosen to allow the ATM/C controller to review the traffic situation and possibly modify the command before validation, thus providing another check against system malfunction. When the command has been validated by the ATM/C controller the command is transmitted to the aircraft via data link, if such is used, at the proper time of execution or the software provides display cues for the ATM/C controller so that he can correctly time the command transmission by voice.

3.6.3 Traffic Display Driver

This function is responsible for maintaining the information displayed on the TD screen. The displayed information and capabilities of current ARTS III displays was chosen as a basis for the design of the

simulation's Traffic Display. The primary goal however was not to duplicate the full spectrum of ARTS III capabilities. Instead the aim was to create an infrastructure that will easily allow incorporation of new functions and visual aids that are needed by the ATM/C controller in order to perform his new role in the automated ground control environment. Figure 3-8 is a blown-up negative of a photograph of the MEGATEK display screen. The displayed items include:

1. Aircraft symbol and tag

The same symbol, a slanted line in the current configuration, is used to represent all aircraft in the ATM/C system. Aircraft under the direct control of the ATM/C supervisor are represented by a backward slanting line (\) while uncontrolled traffic is represented by a forward slanting line (/) for easy identification. The aircraft position is centered around the estimated (surveillance) position of the aircraft target. The aircraft tag is positioned relative to the center of the aircraft symbol. The orientation of the aircraft tag can be selected by the ATM/C controller. Eight possible orientations are available. The following information is included in the aircraft tag:

- a. Flight identification code (transponder code or other appropriate identification used in pilot-controller communications.

- b. Indicated altitude (in 100's of feet)
- c. Latest altitude clearance (in 100's of feet)
- d. Vertical speed symbol (+ if the aircraft is climbing and - if it is descending.)
- e. Ground speed (in knots)

In addition a general purpose message area (five characters) is provided on the aircraft tag. Normally this area displays the aircraft computer identification number. If the aircraft is in some unusual state, however, (e.g. hand-off/hand-over, hazard or conformance alert, etc.) an appropriate blinking message is displayed in this area.

2. Terminal area network

The terminal area network consists of "nodes" and a set of arcs connecting them. It defines nominal arrival and departure air routes within the displayed terminal area airspace. Each type of node is represented by a distinct symbol. Two types of nodes, VORTACs and airway intersection (waypoint), are currently defined. Fixed ground obstructions (hills, radio antennas, etc.) can also be represented as nodes if their existence and position is available. VORTACs and waypoints are accompanied by their name code while for obstructions the minimum safe altitude is displayed.

3. Range rings

Three concentric range rings are included in the display. They are centered around the location of the airfield whose TCA is simulated. They are used to provide the controller with a measure of distances.

4. Airfield layout

A rough sketch depicting the runway layout of the primary airfield as well as other airfields in the simulated area is displayed.

5. Simulation clock

The current simulation time in hours, minutes, and seconds is displayed.

Each item on the screen is displayed with different intensity depending its relative importance. Aircraft symbols and tags are displayed with maximum intensity while range rings are displayed with minimum beam intensity. A medium setting is used for the simulation clock, the terminal area network and the airfield layouts. In addition to the standard display items, special purpose displays can be overlaid on the screen. Figure 3-8 shows such a special purpose display. The operation of this display is presented in chapter 4.

3.6.4 Input Editor

The editing function is responsible for managing the accumulation and display of ATM/C controller keyboard inputs on a character by character basis. Three types of keys are defined. Self-insert keys transmit printable character codes that are inserted in the input buffer and displayed on the display screen. Control keys transmit non-printable characters which are generally not stored in the input buffer. Instead they cause simple editing functions to be performed on the input line. Finally quick action keys are single key commands and are passed immediately to the input identification function for further processing. The processing of quick action keys is independent of the processing of normal input lines which allows some actions to be taken without destroying the normal input already in the input buffer.

3.6.5 Input Processor

The input processor is the heart of the man-machine interface of the simulation facility. It is triggered whenever a command line is available for parsing. The input processor is responsible for parsing the command line, and identifying and performing the requested function.

Parsing the normal input lines resembles syntax analysis in the linguistic sense of the word and is usually more difficult if the command language is flexible enough to provide a convenient and reliable tool of communication between the ATM/C controller and the software.

The currently implemented parser has the following features which enhance its usability:

1. Abbreviation of verbs and keywords
2. Default settings for required but missing keywords
3. Implicit specification of missing parameters
4. Checks for input consistency

The first three enhance the ease of communications between the ATM/C controller and the software while they also increase the probability of error. Abbreviating verbs and keywords for example make it more probable that a misspelled command will be understood by the system as having totally different meaning. In most cases however, erroneous commands result in inputs that are inconsistent among them. As an example, a clearance for an aircraft to climb to 10,000 feet while it is now at 20,000 indicates that something is wrong, though it is not by any means clear what is wrong. Probably the wrong aircraft was addressed. It is however possible that the ATM/C controller meant descend instead of climb or that the ATM/C controller has wrong information on the aircraft's current altitude. A variety of such checks for data consistency have been included. The subject is however vast and deserves extensive study in future research regarding the design of the man-machine interface of the automated ATM/C system.

3.6.6 CTID Update

The CTID update function is triggered after some data on the Controller Tabular Information Display has been modified. Flagging of the modified data is the responsibility of the function that performed the modification.

The Controller Tabular Information Display uses the VT-100 video screen to provide the ATM/C controller with alphanumeric data on the system status and performance. The use of a secondary display is a significant departure from current practice which uses the TD to also display supplementary information. The argument against two separate displays has been that the ATM/C controller can look at the supplementary data without being distracted from his primary task, that of monitoring of target movements on the TD. Minimizing the distraction from the ATM/C controller's primary task is of course desirable. It is not clear however that use of a secondary display will result in greater distraction if it is directly adjacent to the TD screen. At the same time little emphasis has been given to the distraction resulting from the additional effort required to access supplementary information. The second display will provide more display area so that more data can be visible at one time and thus minimize accessing time and effort. The current CTID configuration is shown in figure 3-9. The screen is divided into four sections each displaying a different type of information. The following types of information are currently displayed:

FLIGHT ID	X-PRD	EQ TYPE	NAV CAP	CURRENT ETA	PREFRD ETA	DELAY	SHIFT	CURRENT TRACK	COMMENTS	
EA271										
AA613										
AA864										
Average delay: 1.50 minutes				Runway in use 22L				Barometer setting 29.95		
Maximum delay: 4.75 minutes				Wind from 200 at 10 knots				Ceiling at 3000 feet		
Average shifts: 2				Current objective: MINIMIZE AVERAGE DELAY				Visibility 3 NM		
EA271				TL TO MANJO DIRECT; SPEED 200; DESCEND TO 60				20:09:00		
AA613				TL FLY 40 FROM HTM; SPEED 180; DESCEND TO 50				20:09:30		
AA864				TR HDG 90 ; SPEED 160; DESCEND TO 40				20:13:00		

Figure 3-9. Controller Tabular Information Display Layout

1. Airport scheduling data
2. Simulation statistics
3. ATC messages
4. Computer generated commands

When there is more data of some type than there is space in the screen section on which the data is displayed, the ATM/C controller can scroll the data within the display section using quick action keys defined for this purpose. Currently each type of data occupies a different screen section. Several types of data can share a screen section however. In that case the ATM/C controller will be able to chose which data type he wants displayed at any one time.

3.7 Implementation Status

TASIM is currently fully operational for simulations of the conventional terminal area ATM/C system. The only conventional function which has not yet been implemented is tracking of radar targets. Of the ATM/C automation software, runway scheduling is the only function that has already been implemented. Development of the Nominal and Flight Plan Generators, of the Command Generator, and of the Conformance Monitor software is, therefore, required before experimentation and testing of the fully automated ATM/C system can begin. TASIM has thus far undergone approximately 60 hours of debugging runs and its performance has been satisfactory. To date, however, there has been no systematic testing of all its functions.

CHAPTER 4

THE RUNWAY SCHEDULING PROBLEM

4.1 Introduction

The runway scheduling problem addresses the following question: Given a certain configuration consisting of N_r runways, and the current position of a set of N_a aircraft wishing to land or takeoff, assign to each aircraft i , a runway n and the scheduled time of arrival at that runway, $STAR_{in}^1$, such that some objective function, or system efficiency¹ criterion, is optimized. The schedules are constrained by a wide spectrum of operational and safety requirements. As Dear has shown, there can be substantial improvements in runway capacity and aircraft delays when runway scheduling is applied. [DEA 76]

Scheduling improves runway efficiency by taking advantage of the variation in the minimum time interval allowed by ATC separation standards between pairs of aircraft. These intervals are based on the ATC safety requirements, and vary depending on the type of aircraft and the type of operations involved (arrival-arrival, arrival-departure, etc.). Methods for determining the minimum time intervals between operations have been developed in connection with runway capacity

(1) The term efficiency is used to stress the fact that runway capacity maximization is not always the main objective. Runway capacity can be traded off against reductions in aircraft delay, as well as against more equitable distribution of delays or other user costs.

models, ([PAR 81a], [HOC 74]). Appendix A illustrates how scheduling affects aircraft delays through an example involving aircraft of varying landing speeds.

A substantial improvement can also be realized through coordination of scheduling decisions for both landings and takeoffs on a tactical basis. In current practice, departures are inserted only when gaps in the landing stream allow it. If the departure queue increases beyond some critical length, due to unavailability of adequate gaps in the landing stream, gaps are created by tower controller request until the departure queue dissipates. As is the case with all procedural solutions to operational problems, this method is designed to be easily implementable, but not necessarily efficient. Much idle runway time which seems inevitable today can be eliminated through scheduling the usage of the runway system.

Runway scheduling is tactical. The schedules are generated based on the known traffic at any given time. This means that the runway schedule will have to be updated every time a new aircraft makes its intention to land or takeoff known to the ATM/C system. Given a set of known aircraft, the possible formulations of the problem fall into two distinct categories:

The static formulation assumes that every aircraft in the system is capable of arriving at the runway threshold (for takeoff or landing) at or after some reference time t_0 . Stated differently, this implies that

the current position of aircraft in the terminal area (or the gates) is of no importance to the scheduling problem. All STAR's will be feasible as long as they are greater than t_0 . As a consequence, with the exception of the aircraft chosen to operate first, the STAR's are constrained only by the minimum separation requirements. This assumption greatly simplifies the problem since it allows some formulations in which time is not an explicit variable, but is implicitly accounted for through minimum time intervals between consecutive operations.

The dynamic formulation addresses runway scheduling within the broad framework of ATM/C in the terminal area. It recognizes that the above assumption is weak since the position (within the terminal area airspace or at the aprons) and the operational characteristics of each aircraft affect the earliest time that it can reach the runway threshold. As a result each aircraft's STAR is constrained by the position of the aircraft at the time rescheduling occurs.

4.2 Definitions

This section gives formal definitions for terms and variables that are extensively used throughout the chapter. Variables not appearing here are the ones applicable only to specific sections. Those will be defined as they are needed.

We will generally assume that there are N_a aircraft to be scheduled on N_r runways. When aircraft are classified into types, N_t will denote the number of distinct aircraft types.

Lower case letters $i, j, k, l, m,$ and n will be used as indices. Usually i and j will refer to specific aircraft, m and n will index runways, and k and l will be used to index aircraft types.

The set of aircraft for which scheduling decisions have to be made is called the decision aircraft set, A^d , and is defined as:

$$A^d = \{1, 2, \dots, N_a\}$$

The set of active runways is denoted by R and defined as:

$$R = \{1, 2, \dots, N_r\}$$

Two functions, which provide the correspondence between indices of aircraft and those of runways and aircraft types, are defined. $RWY(i)$ denotes the index of the runway assigned to aircraft i . Similarly $TYPE(i)$ denotes the index of the aircraft type to which i belongs.

The schedule of runway operations will be denoted by S and is defined as the set

$$S = \{ STAR_{in} : i=1, 2, \dots, N_a, n=1, 2, \dots, N_r \} \quad 4-1$$

where, as before, $STAR_{in}$ is the time aircraft i is scheduled to arrive at runway n . $STAR_{in}$ is meaningless if $n \neq RWY(i)$ and it will be set to zero by convention.

In the mathematical formulation of the runway scheduling problem the sequence of operations and the assignment of aircraft to runways will be specified through integer 0-1 variables e_{ijnm} . These are defined as follows:

$$e_{ijnm} = \left. \begin{array}{l} 1 \text{ if } n = RWY(i), \text{ and } m = RWY(j) \\ \text{and } STAR_{in} < STAR_{jm} \\ 1 \text{ if } i = j \text{ and } n = m = RWY(i) \\ 0 \text{ otherwise} \end{array} \right\} \quad 4-2$$

i.e. for two different aircraft i and j , if $e_{ijnm}=1$, aircraft i is scheduled to operate before aircraft j .

Under a given set of separation standards, the minimum time interval allowed between two aircraft i and j assigned to runways n and m respectively will be denoted by s_{ijnm} . The subscripts used stress the dependence of the minimum time separations on the aircraft pair i and j , as well as the runway on which each aircraft operates. Note that the two operations do not have to be in direct succession. The only requirement for s_{ijnm} to be applicable is that aircraft i precedes aircraft j , i.e. $e_{ijnm}=1$. By convention, s_{ijnm} is set to a large negative number when runways n and m are independent.

The lead time constraints, discussed in chapter 2, section 2.2.2, imply that there will be a set of aircraft, denoted by A^0 , whose scheduled time of arrival at the runway or their runway assignment cannot be revised. Such aircraft are not included in the decision aircraft set, A^d . However, they still have to be taken into account in the new schedule since they affect the schedules of other aircraft which are still eligible for rescheduling. The runway schedule, therefore, is optimized over all aircraft in the set A^d , while aircraft in the set A^0 appear in the constraints to insure that proper separations are maintained. The set A^+ of all aircraft in the terminal area ATM/C system can now be defined as:

$$A^+ = A^0 \cup A^d$$

The concept of the Earliest and Latest Feasible Time of Arrival at the Runway was introduced in chapter 2. We now formalize their definitions by making explicit their dependence on the runway, as well as the aircraft. We will denote them by $EFTAR_{in}$ and $LFTAR_{in}$. $EFTAR_{in}$ and $LFTAR_{in}$ depend on the aircraft characteristics (primarily its airspeed) and represent the time interval within which each aircraft can reach the runway threshold from its current position. For landing aircraft, this interval reflects the flexibility provided by the TFPG in expediting the aircraft through the terminal area and in absorbing delays imposed by the schedule. Note that, if holding is allowed at the entry fixes, $LFTAR_{in}$ for landing aircraft will not be limited by flight planning considerations before the aircraft reach the entry fix or while

they are holding there. At that stage of the approach, fuel availability onboard the aircraft is the limiting factor for $LFTAR_{in}$. For takeoffs, $EFTAR_{in}$ reflects the taxi time to runway n , while $LFTAR_{in}$ will typically be very large since there need not be any restriction to how long aircraft may be on gate hold.

When $LFTAR_{in}$ is very large, it is possible that, given the proper conditions, aircraft i will be pushed at the end of the sequence every time the runway schedule is revised. In order to avoid such a possibility, the position an aircraft can occupy in the sequence of operations will need to be constrained. We will denote by \min^{POS}_{in} and \max^{POS}_{in} , the minimum and maximum positions that aircraft i can occupy in the sequence within a certain runway n . The limits applicable to the position of aircraft i in the overall sequence will be denoted by \min^{POS}_i and \max^{POS}_i .

Finally, in the mixed integer linear programming formulation of the next section, we will use the symbol M to denote a very large positive constant.

4.3 Formulation of the Dynamic Runway Scheduling Problem

We are now in the position to formulate the dynamic runway scheduling problem as a mixed integer mathematical program.

Find $STAR_{in}$ and e_{ijnm} , for $i, j \in A^+$ and $n, m \in R$, that minimize some specified objective function $Z(S)$ and satisfy the following constraints.

1. Scheduling Constraints

$$\begin{aligned}
 \text{a) } \quad & \text{STAR}_{jm} \geq \text{EFTAR}_{jm} \cdot e_{jjmm} && \forall j \in A^d \\
 & && \forall m \in R \\
 \text{b) } \quad & \text{STAR}_{jm} \leq \text{LFTAR}_{jm} \cdot e_{jjmm}
 \end{aligned}$$

2. Sequencing Constraints

$$\begin{aligned}
 \text{a) } \quad & \sum_{i \in A^+}^{i \neq j} e_{ijmm} \geq \min \text{POS}_{jm} \cdot e_{jjmm} && \forall j \in A^d \\
 & && \forall m \in R
 \end{aligned}$$

$$\begin{aligned}
 \text{b) } \quad & \sum_{i \in A^+}^{i \neq j} e_{ijmm} \leq \max \text{POS}_{jm} \cdot e_{jjmm} \\
 & && \forall j \in A^d \\
 & && \forall m \in R
 \end{aligned}$$

$$\begin{aligned}
 \text{c) } \quad & \sum_{n \in R} \sum_{m \in R} \sum_{i \in A^+}^{i \neq j} e_{ijnm} \geq \min \text{POS}_j && \forall j \in A^d
 \end{aligned}$$

$$\begin{aligned}
 \text{d) } \quad & \sum_{n \in R} \sum_{m \in R} \sum_{i \in A^+}^{i \neq j} e_{ijnm} \leq \max \text{POS}_j && \forall j \in A^d
 \end{aligned}$$

3. Spacing Constraints

$$\text{STAR}_{jm} \geq \text{STAR}_{in} + s_{ijnm} \cdot e_{ijnm} - (1 - e_{ijnm}) \cdot M$$

$$\forall i \in A^+ \text{ and } j \in A^+$$

$$\forall n \in R \text{ and } m \in R$$

4. Assignment Constraints

a) $e_{ijnm} \in \{0,1\}$ $\forall i \in A^+, j \in A^+$
 $\forall n \in R, m \in R$

b) $\sum_{m \in R} e_{jjmm} = 1$ $\forall j \in A^d$

c) $\sum_{n \in R} \sum_{m \in R} (e_{ijnm} + e_{jimn}) = 1$ $\forall j \in A^d$
 $\forall i \in A^+ - \{j\}$

d) $\sum_{n \in R} \sum_{i \in A^+}^{i \neq j} e_{ijnm} \leq M \cdot e_{jjmm}$ $\forall j \in A^d, \forall m \in R$

e) $\sum_{n \in R} \sum_{i \in A^+}^{i \neq j} e_{jimn} \leq M \cdot e_{jjmm}$ $\forall j \in A^d, \forall m \in R$

The scheduling constraints (1a and 1b) simply require that, for all aircraft j , STAR_{jm} lies in the interval $[\text{EFTAR}_{jm}, \text{LFTAR}_{jm}]$ when aircraft j is assigned to runway m (i.e. if $e_{jjmm} = 1$). When aircraft j is not

assigned to runway m , the two constraints force $STAR_{jm}$ to be zero, which is consistent with the convention adopted for that case.

The sequencing constraints operate similarly. In constraints 2a and 2b, the summation of e_{ijmm} over all aircraft i , is the number of aircraft assigned to operate on runway m prior to aircraft j , since e_{ijmm} is zero unless both i and j are assigned to runway m and i is scheduled to operate before j . When j is not assigned to runway m , the right hand side of the constraint becomes zero. Consequently, all e_{ijmm} are constrained to be zero. Constraints 2c and 2d are equivalent to 2a and 2b except that they apply to the position of aircraft j in the overall sequence.

In the spacing constraints, M is assumed to be a large constant compared to the problem variables and parameters. When e_{ijnm} is equal to 1, i.e. when aircraft i is assigned to operate on runway n before aircraft j operates on its assigned runway m , the constraint insures that $STAR_{jm}$ is greater than $STAR_{in}$ by at least s_{ijnm} , the required minimum time separation between the two operations. In all other cases e_{ijnm} is zero and the constraint is redundant, since the right hand side is dominated by M .

Finally, the assignment constraints insure that the values of the sequencing variables are consistent. Constraint 4a limits the sequencing variables to the two integer values 0 and 1. Constraint 4b requires that every aircraft is assigned to a runway once and only once.

Constraint 4c insures that either aircraft i is scheduled after aircraft j , or j is scheduled after i , but not both. Finally, constraints 4d and 4e state that if e_{jjmm} is zero, e_{ijnm} and e_{jimn} should also be zero for all aircraft i and runways n . This insures that each aircraft is consistently assigned to one runway.

The objective function in the above formulation is a general function of the schedule S (i.e. of the $STAR_{in}$'s). The next section examines runway efficiency measures and develops alternative objective functions to be optimized by runway scheduling.

4.4 Measures of Runway Efficiency

Runway efficiency is a multi-dimensional quantity. It is impossible to capture every aspect of the operation of the runway system in a single number. We, therefore, use several measures which, taken collectively, describe the term. In this work we will consider the following measures of runway efficiency:

1. Runway capacity
2. Aircraft delay (total or average)
3. Weighted aircraft delay (total or average)
4. Equitable distribution of delay or fuel costs

The mathematical expressions which are developed in the next several sections for each of these measures, are functions of the time

each aircraft reaches the runway. We will refer to these expressions interchangeably as cost functions or objective functions.

4.4.1 Runway Capacity

Runway capacity is the most widely used runway efficiency measure. The normal use of the term refers to the saturation capacity of the runway system. Saturation capacity is defined as the average number of aircraft that can takeoff and/or land at the runway system during some unit of time, assuming infinite demand. A number of models have been developed to estimate analytically the saturation capacity for a variety of runway configurations.¹ These models use the time intervals between operations to calculate the mean time interval between successive operations, $\overline{\Delta t}$. The saturation capacity is then simply the inverse of $\overline{\Delta t}$, i.e.,

$$C_{\text{sat}} = 1 / \overline{\Delta t} \quad 4-3$$

$\overline{\Delta t}$ is averaged over all successive pairs of aircraft for a specific sequence, and over all possible sequences. The contribution of each sequence is weighted by its probability of occurrence given some mix of aircraft types.

The saturation capacity, as quantified above, is an expected value based on a representative aircraft mix. It is, therefore an

(1) See for example [PAR 81a] and [HOC 74].

inappropriate measure for our purposes. We are interested in optimizing the capacity of the runway system at a particular instant in time, i.e. based on a specific "snapshot" of the traffic wishing to land and takeoff. This capacity will generally be substantially different from the capacity calculated in equation 4-3. Naturally, in the absence of lead time constraints or other operational restrictions, by optimizing the "snapshot" capacity repeatedly as the traffic situation changes, we also optimize the long term capacity of the runway system.

Consider some schedule S. The average interval between successive operations achieved by this schedule is given by the difference between the scheduled time of the last operation and that of the first operation, divided by the total number of operations, i.e. the cardinality of the set A^+ .

$$\Delta t(S) = \left(\max_{\substack{i \in A^+ \\ n \in R}} \{STAR_{in}\} - \min_{\substack{i \in A^+ \\ n \in R}} \{STAR_{in}\} \right) / \|A^+\| \quad 4-4$$

The runway system throughput rate can now be defined as the inverse of the average interval between successive operations:

$$TR(S) = 1 / \Delta t(S) \quad 4-5$$

Throughput rate is maximized if the interval between the first and the last operation is minimized. The last (or maximum) scheduled time (MST) can thus be used as the objective function for the optimization. Formally the objective function can be written as:

$$\begin{aligned} Z(S) &= \text{MST}(S) \\ &= \max_{\substack{i \in A^+ \\ n \in R}} \{ \text{STAR}_{in} \} \end{aligned} \quad 4-6$$

and the optimization of section 4.3 is a minimax problem since the objective becomes:

$$\min_{\substack{i \in A^+ \\ n \in R}} Z(S) = \min_{\substack{i \in A^+ \\ n \in R}} \max_{\substack{i \in A^+ \\ n \in R}} \{ \text{STAR}_{in} \}$$

The new objective function can be easily converted back into a normal minimization by a standard transformation, i.e. by defining a new decision variable t_{\max} , by substituting

$$\min t_{\max}$$

for the above objective and by adding the following set of constraints:

$$\text{STAR}_{jm} \leq t_{\max} \quad \forall j \in A^+, \forall m \in R$$

4.4.2 Aircraft Delays

Delays experienced by aircraft within the terminal area ATM/C system are responsible for a large fraction of airline costs. The direct operating cost (fuel, crew salaries, etc.) is increased due to the increase in the block time for the trip. The indirect operating cost is also increased due to the lower aircraft utilization rate (e.g. passenger miles per unit of time). The cost of the trip as perceived by

the passenger is also increased since delays effectively decrease the average speed of air travel. These effects on airlines and passengers alike are particularly obvious on short trips.

The delay experienced by an aircraft in the terminal area is the difference between the actual time the aircraft arrives at the assigned runway and the time it would have arrived in the absence of any other traffic. The latter is called the preferred time of arrival at the runway, $PTAR_i$. We note that the runway assigned to the aircraft in the absence of other traffic will be the one the aircraft can safely reach first. So if we let $PTAR_{in}$ be the preferred time of arrival at runway n we have:

$$PTAR_i = \min_{\substack{d \\ i \in A \\ n \in R}} \{ PTAR_{in} \}$$

For planning purposes we will use $STAR_{in}$ instead of the actual time of arrival at the runway which, of course, is not known.

The aircraft delay is given by:

$$d_i = \sum_{n \in R} STAR_{in} - PTAR_i$$

or

$$d_i = STAR_{i, RWY(i)} - PTAR_i$$

since $STAR_{in}$ is non-zero only for $n = RWY(i)$.

Several delay related objective functions can be of interest.

The total aircraft delay, TAD(S), is simply given by summing the delays experienced by all the aircraft in the system. For some runway schedule S therefore,

$$TAD(S) = \sum_{i=1}^{N_a} d_i$$

The average aircraft delay is given by:

$$AD(S) = TAD(S) / N_a$$

For any instance of the runway scheduling problem, N_a is a constant. The total and average aircraft delay are therefore equivalent objectives, since the schedule that minimizes one will also minimize the other. Furthermore since

$$TAD(S) = \sum_{i=1}^{N_a} (STAR_{i,RWY(i)} - PTAR_i)$$

$$= \sum_{i=1}^{N_a} STAR_{i,RWY(i)} - \sum_{i=1}^{N_a} PTAR_i$$

4-7

and the second term is a constant, minimization of average aircraft delay is equivalent to minimizing the sum of the STAR's.

The objective functions considered thus far implicitly assume that the cost associated with aircraft delays is the same for all aircraft.

This is not generally true. The total weighted delay allows the importance attached to the unit time of delay experienced by different aircraft to vary. We define

$$\text{TWD}(S) = \sum_{i=1}^{N_a} w_i d_i$$

where w_i is the relative importance, or weight, of the unit time of delay experienced by aircraft i .

The weights will typically be representative of the costs associated with a unit time of delay (e.g. the direct operating cost), and the same weight will be applied to all aircraft of the same type, i.e.

$$w_i = w_j \quad \text{if TYPE}(i) = \text{TYPE}(j)$$

This assumption is necessary for some algorithms presented in this chapter. In general however, each aircraft may be assigned a distinct weight.

The average weighted delay is defined as:

$$\text{AWD}(S) = \text{TWD}(S) / \left(\sum_{i=1}^{N_a} w_i \right)$$

Since the denominator is again a constant for each instance of the runway scheduling problem, the two objective functions are equivalent. Similarly, the weighted sum of the STAR's,

$$\begin{aligned} \text{WSST}(S) &= \sum_{i=1}^{N_a} w_i \left(\sum_{n=1}^{N_r} \text{STAR}_{i,n} \right) \\ &= \sum_{i=1}^{N_a} w_i \text{STAR}_{i, \text{RWY}(i)} \end{aligned} \quad 4-8$$

is also equivalent to TWD(S) and to AWD(S) and will be the general objective function used when discussing optimization of aircraft delays.

We now turn to an even more general case of delay related objective functions, where the contribution of each aircraft is additive, but each contribution is not necessarily a linear function of the delay. We will refer to this class of objective as Generalized Weighted Delays.

$$\text{GWD}(S) = \sum_{i=1}^{N_a} f_i(d_i) \quad 4-9$$

Note that f_i may be distinct for each aircraft, and that TWD(S) is a special case of GWD(S) for which f_i 's are linear functions of the delays, i.e.

$$f_i = w_i d_i + (\text{constant})$$

4.4.3 User Cost Distribution: The CPS Methodology

We finally turn to the issue of delay (or other cost) distribution among the users, or classes of users,¹ of the terminal area ATM/C system. Inequities in the distribution of user costs almost always arise when system-wide optimization of multi-user systems is performed. Such inequities will not be tolerated by the users, and it is essential that they are dealt with by the optimization process.

Objective functions can be developed to remedy this situation. If the maximum user cost (instead of the average cost) is used as the optimization criterion, for example, the optimum schedule will be such that all user costs will tend to be roughly the same. The drawback with this approach is that such optimization criteria are generally too weak to result in significant improvements on a system-wide basis.

A preferable method is to impose constraints on the runway schedules which insure the non-preferential treatment of all system users. A wide variety of explicit constraints can be imposed to achieve this goal. For example, the average cost experienced by each user class can be constrained not to exceed a certain percentage of the average cost experienced by all the system users collectively. This type of constraint will alleviate large inequities but there is no guarantee

(1) Users may be classified in several different ways. One obvious classification may be by aircraft type, while in discussing cost distribution, it may be more appropriate to put all aircraft of each airline in a distinct user class.

that some user class will not be consistently discriminated against. The reason is that they do not remove the cause of the inequities. They merely limit their effect.

Inequities in the runway scheduling problem arise because runway efficiency is best maximized by "bunching" of aircraft with similar characteristics. This is particularly evident in the case of landing aircraft where those of the same landing speed should be bunched for maximum efficiency.¹ Departing aircraft behave similarly with respect to the departure route assigned to them and to the runway on which they are to take off. As a consequence, "minority" aircraft, i.e. those that are dissimilar to most of the traffic at any given time, will tend to be pushed back in the schedule.

The phenomenon of "bunching" suggests that inequities can be best avoided by restricting the position of the aircraft in the sequence rather than by restricting the amount of delay each group of aircraft experiences. This work will, therefore, concentrate on a class of constraints which limit the position of the aircraft in the sequence. The Constrained Position Shifting (CPS) methodology, proposed by Dear, is designed to achieve this goal [DEA 76].

CPS seeks to limit the forward and backward movement of all aircraft in the ATM/C system. The nominal sequence of operations (as

(1) See section 4.6 for further discussion of this effect. Also [DEA 76] and [PSA 78] include extensive analysis of aircraft bunching for the case of scheduling landings on a single runway.

determined by the relative PTAR's) is used as the basis for the constraints imposed on the position of aircraft in the optimal schedule. The Maximum number of Position Shifts (MPS) is the largest deviation from the nominal position that will be allowed for any aircraft. MPS can be arbitrarily set to any value. For an MPS of zero only the nominal sequence of operations is feasible and no optimization is achieved. As MPS increases from 0 to N_a , better and better (from the point of view of the value of the objective function) schedules can be determined. When MPS is set to N_a the position of all aircraft in the sequence is unconstrained. Experience has shown that MPS values of 4 or 5 achieve values for the objective function that are very close to those without any position shifting constraints. At the same time it insures non-preferential treatment of all aircraft in the ATM/C system.¹

The sequencing constraints in the formulation of section 4.3 implement the CPS methodology. Note that the maximum and minimum allowable position in the sequence for each aircraft ($\max \text{POS}_{in}$ and $\min \text{POS}_{in}$) are expressed in absolute terms. One could let

$$\max \text{POS}_{in} = \text{POS}_{nom}(i) + \text{MPS}$$

and

$$\min \text{POS}_{in} = \text{POS}_{nom}(i) - \text{MPS}$$

(1) See for example [DEA 76] and [PSA 78]

to make the relationship of the limits on the position of aircraft i to its position in the nominal sequence, $POS_{nom}(i)$, explicit. The absolute form of the sequencing constraints allows the CPS methodology to be used as a means of imposing a variety of operational constraints to the schedule. For example, we can have direct control of the position of some aircraft by setting the minimum and maximum positions for that aircraft to the same value. Similarly, we can force two aircraft to operate in direct succession if needed. Such options can and will be made available to the ATM/C controller, as a means of controlling the scheduling function of the automation software.

4.5 The Complexity of the Runway Scheduling Problem

For each problem we can identify a number which is representative of the quantity of input variables defining it. This number is called the size of the problem. The size of the runway scheduling problem may be the number of 0-1 variables in the formulation of section 4.3 i.e.,

$$SIZE(RSP) = (N_a N_r)^2$$

The time complexity, $T(n)$, of an algorithm that solves some problem of size n , is defined as some function $f(n)$ such that: the number of time units required to obtain a solution is equal to $Cf(n)$, for some constant C . We then say that the time complexity of the algorithm is of the order of $f(n)$, and write:

$$T(n) = O(f(n))$$

Even though the time complexity is a property of algorithms rather than of the problem itself, we can talk about the time complexity of a problem if that is understood to mean the time complexity of the most efficient algorithm that can solve the problem.

In order to determine the time complexity of the runway scheduling problem, we note that the set of constraints that are imposed on the schedules are all linear functions of the problem variables, $STAR_{in}$ and e_{ijnm} . Consequently, the formulation is a mixed integer linear program (MILP) as long as the objective function is also a linear function of the problem variables. This is obviously the case when $WSST(S)$ is minimized. When $MST(S)$ is the objective function, a standard transformation can be used to obtain an equivalent MILP as shown in section 4.4.1.

In the case of generalized weighted delays there is no exact method which will transform the problem into the standard MILP formulation. We can however approximate this problem as an MILP if we substitute each weight function by a piece-wise linear approximation (see [BRA 77]). Furthermore, if all the generalized weights, $f_i(d_i)$ are convex functions of the aircraft delays d_i , the approximation does not increase the complexity of the problem.

The time complexity of the best known general purpose algorithm (e.g. Branch and Bound) that solves the runway scheduling problem is $O(\exp[(N_a N_r)^2])$. This value can be improved by recognizing that, due to

the assignment constraints of section 4.3, the selection of the e_{ijnm} 's is not independent. For example, if e_{iinn} is set to 1 for some value of n , then e_{iimm} has to be zero for all $m \neq n$. We will show that the time complexity of the problem is exponential with respect to the number of aircraft in the system and polynomial with respect to the number of active runways.

To make the discussion concrete, a simple algorithm that solves the problem will be analyzed. It consists of generating the feasible sets of values for e_{ijnm} 's and solving the remaining linear program for each such set.¹ Obviously the algorithm will have to generate only feasible sets of values. Generating all possible sets of values and then using the assignment constraints to reject the ones that are infeasible will not be an improvement over the classical branch and bound approach.

Feasible sets of values can be generated efficiently by first assigning aircraft to runways, i.e. selecting values for e_{iinn} 's, and then generating all the possible sequences within each runway. From the aircraft sequences the rest of the values can be generated in polynomial time.

The number of ways aircraft can be assigned to runways consistently is $(N_r)^a$ since each aircraft can be assigned to any runway (i.e. N_r alternative ways) independently of any other aircraft assignment.

(1) Note that, with the exception of the case where $GWD(S)$ is the objective function, the solution to the resulting LP is trivial.

Furthermore, for a specific set of runway assignments, the number of distinct sequences that can be constructed are:

$$\prod_{n=1}^{N_r} K_n !$$

where, K_n is the number of aircraft assigned to runway n . This number is of the order of $N_a !$ since it is (at least in principle) possible to assign all aircraft to one runway leaving all the others idle.

The time complexity of the runway scheduling problem is therefore,

$$T(\text{RSP}) = O((N_r)^{N_a} N_a !)$$

Instead of generating all sequences within each runway we may use the sequencing constraints to further limit the number of distinct sets of values that have to be evaluated. Suppose that each aircraft is only allowed to be shifted by at most MPS (forward or backward) from its nominal position. Let us consider the following recursive algorithm for generating sequences within each runway:

Step 0: Determine the next aircraft in the nominal sequence that has not been assigned a position. If all aircraft have been assigned positions go to step 3. Otherwise remove this aircraft from the list and continue with step 1.

Step 1: Scan the list of allowable positions for this aircraft.

If the list is empty, go to step 1a. Otherwise go to

step 1b.

Step 1a: Reset the list of allowable positions for this aircraft.

Reenter the aircraft to the list of aircraft without position assignment. Return to the previous calling level of the algorithm. If this is the top calling level then STOP.

Step 1b: Assign to the aircraft a position from the list that has not been already assigned to another aircraft and delete the position from the list. If all the allowable positions have already been assigned go to step 1a.

Step 2: Apply this algorithm on the aircraft that remain without an assigned position. Upon return from the next calling level go to step 1.

Step 3: Generate the e_{ijmm} 's from the sequence. Evaluate the objective function and return to the previous level of the algorithm.

This algorithm in essence performs "depth-first"⁽¹⁾ traversal of a tree. The nodes on the k^{th} level of the tree represent all the possible position assignments to $(k-1)$ aircraft. Accordingly the depth of the tree for runway n is $K_n + 1$, where, as before, K_n is the number of aircraft assigned to the runway. The number of branches from each node on the k^{th} level correspond to the number of allowable positions for the k^{th} aircraft.

(1) The term depth-first means that the branches out of any node are traversed before any additional nodes on the same level are reached.

In order to determine the number of branches from each node, let us assume that at the k^{th} level we are considering the aircraft i for which $\text{POS}_{\text{nom}}(i)=k$. From the root of the tree there will be exactly $(\text{MPS}+1)$ branches since the first aircraft will be allowed to occupy positions 1, 2, ..., MPS , $\text{MPS}+1$. The second aircraft will be allowed on positions 1, 2, ..., $\text{MPS}+1$, $\text{MPS}+2$. But since one of these positions will be occupied by the first aircraft, there will only be $(\text{MPS}+1)$ feasible positions for this aircraft as well. By the same reasoning, at each level above $(K_n - \text{MPS})$ there will be $(\text{MPS}+1)$ branches from each node. Finally, the number of branches from nodes at the last MPS levels are limited by the number of positions that are not yet occupied as opposed to the number of allowable positions.

Consequently the total number of leaves on the tree, and therefore the total number of feasible sequences that need to be evaluated is given by:

$$(\text{MPS})! (\text{MPS}+1)^{(K_n - \text{MPS})}$$

Each step of the algorithm can be performed in polynomial time. Furthermore, each leaf of the tree requires the recursive part of the algorithm to be performed at most K_n times. The time complexity of the algorithm is therefore dominated by the number of leaves that have to be reached.

Again the worst case is when all aircraft are assigned to one runway. By considering MPS to be a constant the number of leaves is

$$O((MPS+1)^N)$$

We can now combine this result with the number of possible runway assignments to obtain:

$$T(RSP) = O([N_r (MPS+1)]^N) \quad 4-10$$

which is polynomial in terms of the number of runways, N_r , and exponential in terms of the number of aircraft, N_a .

4.6 Variations of the Runway Scheduling Problem

In this section we examine three variations of the runway scheduling problem for which exact algorithms have appeared in the literature. In all cases only landing aircraft are considered. Furthermore, the scheduling constraints of section 4.3 are assumed non-binding. According to the classification of section 4.1, therefore, the variations to be discussed here belong to the class of static runway scheduling problems.

The algorithms will not be described in quantitative terms. Instead we will focus on the assumptions that make the approach taken possible and how the algorithm fails when any of these assumptions is not valid. The goal will be to:

1. Explore the special characteristics of the problem that emerge.
2. Identify desirable properties of the optimal solutions that may be used to advantage in obtaining solutions to the general problem, as well as undesirable ones that require additional constraints to be imposed in order to insure that the resulting schedules are implementable.
3. Gain insight on various aspects of the problem and on how their interactions affect its complexity.

4.6.1 Scheduling Landings on a Single Runway (MPS=infinity)

This is the simplest in the class of runway scheduling problems. It was studied extensively by Dear, [DEA 76]. In our terminology¹ the problem can be stated as follows:

Given N_a aircraft wishing to land on a single runway and assuming that

1. $PTAR_i = EFTAR_i = t_0$, $\forall i=1,2,\dots,N_a$
2. $LFTAR_i = \infty$, $\forall i=1,2,\dots,N_a$
3. $\max POS_i = N_a$, $\forall i=1,2,\dots,N_a$
4. $\min POS_i = 0$, $\forall i=1,2,\dots,N_a$

(1) Runway subscripts are not needed for this discussion and have been suppressed.

find the runway schedule which minimizes the maximum scheduled time of arrival at the runway (MST(S)). Here, t_0 is some reference time which can be arbitrarily set to zero.

Dear proves that the optimal solution to this problem can be analytically derived, and it is unique. The optimal schedule is the one in which each aircraft succeeds all others of lower (or equal) landing speed and precedes all others of higher (or equal) landing speed. Dear calls this an ascending (in terms of landing speeds) sequence. Dear similarly defines a descending sequence as one in which each aircraft succeeds all others of higher landing speed and precedes all others of lower landing speed.

Analytical solutions are also derived if initial and final constraints are imposed on the landing sequence. In particular, it is assumed that there exists one aircraft with fixed landing time $STAR_0 = t_0$, which is constrained to land first and another aircraft, i_f , which is constrained to occupy the last position in the sequence. Of course, the scheduled time of arrival at the runway for i_f is not fixed. The optimal solution in the constrained cases is shown to consist of at most three subsequences, either one ascending and two descending, or two ascending and one descending.

The above results are based on the special structure exhibited by the matrix of time separations between successive landings. These results, therefore, cannot be extended to cases which include departing

aircraft because the arrival-departure separations do not have the same structure.

Dear's results bring out several important aspects of the runway scheduling problem.

1. They clearly demonstrate the phenomenon of "bunching" of aircraft with similar characteristics (i.e. similar landing speeds in this case). Furthermore, the bunching is a direct result of the relationship between landing speeds and minimum aircraft separations.
2. Imposing initial and final constraints on the runway schedules affects the optimal solution in two ways: First, instead of a single ascending sequence, (i.e. a unique solution), we now have a multiplicity of solutions since the actual number of aircraft in each of the three subsequences does not affect the value of the objective function. Second, aircraft bunching is not as pronounced since, even though aircraft of the same landing speed are in direct succession within each subsequence, they can now be in up to three different places in the overall sequence.
3. In every case, the optimal solutions are not defined in terms of aircraft schedules but in terms aircraft sequences. Behind this transformation lies an assumption which allows a unique schedule to be derived from a

sequence of operations. We will call it the scheduling assumption. In this case, the scheduling assumption is simply that the runway will not remain idle unnecessarily. Since no scheduling constraints are imposed, the scheduled time of arrival at the runway for any aircraft j can be determined by:

$$STAR_j = STAR_i + s_{ij}$$

where, i is the aircraft directly preceding j in the sequence. The same scheduling assumption is used in the algorithms which will be examined in the next two subsections. A more general version of this assumption is used in the heuristic algorithm described in section 4.8.

4.6.2 Scheduling Landings on a Single Runway ($MPS < N_a - 1$)

This version of the runway scheduling problem is similar to the one that was described in the previous subsection. Now however, sequencing constraints are imposed on some or all aircraft. The algorithm does not depend on any particular structure of the sequencing constraints. Since it was developed based on the CPS methodology however, we will assume that each aircraft i can be shifted up to MPS positions forward or backward from its (unique) nominal position, $POS_{nom}(i)$. Accordingly the problem can be formulated as follows:

Given N_a aircraft wishing to land on a single runway, and assuming that,

1. $PTAR_i = EFTAR_i = t_0$, $\forall i=1,2,\dots,N_a$
2. $LFTAR_i = \infty$, $\forall i=1,2,\dots,N_a$
3. $\max_i POS_i = POS_{nom}(i) + MPS$, $\forall i=1,2,\dots,N_a$
4. $\min_i POS_i = POS_{nom}(i) - MPS$, $\forall i=1,2,\dots,N_a$

where, $POS_{nom}(i)$ and MPS are constants, find the runway schedule which minimizes the maximum scheduled time of arrival at the runway (MST(S)).

Psaraftis, [PSA 78], developed a dynamic programming algorithm to solve this problem. First he showed that, given the scheduling assumption presented in the previous section and ignoring the sequencing constraints, the problem could be formulated as a classical Travelling Salesman Problem, (TSP). Furthermore, the dynamic programming approach for solving TSP's could be modified to incorporate the sequencing constraints. This was done by letting the value of the objective function go to infinity whenever an infeasible state was reached. The approach can also be modified to handle the total aircraft delay and the weighted sum of the aircraft delays as objectives. The time complexity of this algorithm can be shown to be:

$$T_{DP}(RSP) = O(N_a \cdot 2^{N_a})$$

At this point Psaraftis made a key assumption which drastically reduced the time complexity of the algorithm from an exponential to a

polynomial function of the number of aircraft. He took advantage of the fact that aircraft can be classified into categories (or types) each with similar characteristics. These types can be defined such that, for all $i=1,2,\dots,N_a$

$$\text{TYPE}(j) = \text{TYPE}(k) \text{ iff } \left\{ \begin{array}{l} s_{ij} = s_{ik} \\ \text{and} \\ s_{ji} = s_{ki} \end{array} \right\}$$

i.e. within each category, all aircraft are indistinguishable with respect to their time separations.

This assumption allows a more compact representation of the original state-stage diagram associated with the dynamic programming formulation. The worst case (in terms of time complexity) for the modified formulation occurs when each of the aircraft types contains the same number of aircraft. Letting N_t be the number of aircraft types and denoting by $[x]$ the smallest integer which is greater or equal to x , the time complexity of the modified algorithm is

$$T_{\text{MDP}}(\text{RSP}) = O(N_t [[N_a/N_t] + 1]^{N_t})$$

The time complexity of the modified algorithm is then a polynomial function of the number of aircraft. It remains exponential, however, with respect to the number of aircraft types. As expected, $T_{\text{DP}}(\text{RSP})$ and $T_{\text{MDP}}(\text{RSP})$ are identical when $N_t = N_a$, i.e. when no two aircraft are of the same type.

Unlike the analytical results derived by Dear, the dynamic programming approach does not depend on a particular structure of the aircraft separations. Instead, the only condition necessary for the validity of the optimality recursion is that the separation matrix satisfies the triangle inequality, i.e.

$$s_{ij} + s_{jk} > s_{ik}$$

for all aircraft triplets i , j , and k . When this inequality is satisfied, the information required at each stage is limited to only the last scheduled aircraft as well as the number of aircraft in each aircraft class that have not yet been scheduled.

We can generalize this result by observing that if the triangle inequality is false but, for any four aircraft i , j , k , l , the inequality:

$$s_{ij} + s_{jk} + s_{kl} > s_{il}$$

is satisfied, we can assure correct spacing by maintaining information on the last two scheduled aircraft at every stage of the dynamic program. Similarly, if the equivalent inequality is satisfied for all aircraft n -tuples the state representation of the dynamic program has to maintain information on the last $n-2$ aircraft in order to assure correct separations among all aircraft pairs. In this general case, the time complexity of the algorithm is:

$$T_{MDP}(RSP) = O(N_t^{(n-2)} [(N_a/N_t)+1]^{N_t})$$

For landing aircraft the separation matrix satisfies the triangle inequality. This is not the case, however, when both takeoffs and landings are to be scheduled. In particular, the triangle inequality may not hold if j is a departing aircraft scheduled between two landings, i and k . The typical value of n for mixed operations is 5 or 6. That is, it is possible to insert 2 or 3 departures between some pairs of landing aircraft without increasing the required separations between them.

It is instructive to consider what happens if scheduling constraints are included in the dynamic programming formulation to the runway scheduling problem studied by Psaraftis. At first, it may seem that these can be handled the same way the sequencing constraints are handled. Namely, each state would now have to satisfy two feasibility conditions instead of one. The only additional requirement would be that now the scheduled time of arrival at the runway as well as the value of the objective function associated with each state would have to be stored. Unfortunately, this is the case only when the runway throughput rate is maximized.

When other objective functions are optimized, this approach fails to guarantee an optimal solution because the optimality criterion for the dynamic program is not satisfied. Looking at it from a different perspective, when scheduling constraints are imposed the state-stage description is incomplete unless the time variable is introduced as part

of it. In particular, time has to be the stage variable along with the number of aircraft for which scheduling decisions have been made.

Another, more subtle, problem is that the classification of aircraft into types is no longer valid. In the absence of scheduling constraints, it is reasonable to assume that aircraft of a specific type will land in their relative nominal order. In any case, the relative order in which aircraft of a certain class land has no effect on the objective function. This is no longer true when scheduling constraints are present. In essence, aircraft within each type are no longer indistinguishable since each has distinct limits on its scheduled time of arrival at the runway.

4.6.3 Scheduling Landings on Independent Runways ($MPS < N_a - 1$)

We now turn to the case where two (or more) independent¹ runways are active. In all other respects the problem considered here is very similar to the ones considered in the previous two sections.

Given N_a aircraft wishing to land on two independent runways, and assuming that,

(1) Two runways are independent when aircraft operations on the two runways need not be coordinated. In effect, each runway can be scheduled as if the other did not exist.

1. $PTAR_{in} = EFTAR_{in} = t_o$, $\forall i=1,2,\dots,N_a$
 $\forall n=1,2$
2. $LFTAR_{in} = \infty$, $\forall i=1,2,\dots,N_a$
 $\forall n=1,2$
3. $\max_{in} POS_{in} = POS_{nom}(i) + MPS$, $\forall i=1,2,\dots,N_a$
 $\forall n=1,2$
4. $\min_{in} POS_{in} = POS_{nom}(i) - MPS$, $\forall i=1,2,\dots,N_a$
 $\forall n=1,2$

where, $POS_{nom}(i)$ and MPS are constants, find the runway schedule which minimizes the maximum scheduled time of arrival at the runway (MST(S)).

In the absence of any sequencing constraints, this problem could be formulated as a dynamic program and Psaraftis' algorithm can be extended to solve it [PSA 78]. This approach fails when sequencing constraints are present.

The author developed a Branch and Bound algorithm to solve this problem, [PAR 78]. The 2-tour TSP formulation was used and sequencing constraints were incorporated by introducing artificial nodes in its graph representation.

In the graph representation of the unconstrained problem, each node represents an aircraft to be scheduled. In the modified graph, each aircraft was represented by a number of nodes each corresponding to a possible position and runway assignment for the aircraft. The constrained problem can be shown to be a 2-tour TSP defined on the modified graph. The number of nodes on the new graph is considerably larger than that of the graph of the unconstrained problem. Both

problems, however, are shown to be of the same time complexity, i.e the constrained problem is exponential with respect to the number of aircraft in the system as opposed to the number of nodes in the modified graph.

In addition to having exponential time complexity, this algorithm has the same problems as Psaraftis' dynamic programming approach with respect to scheduling constraints and inclusion of departing aircraft.

4.7 Heuristic Versus Exact Algorithms

The three variations of runway scheduling discussed in the previous section provide good evidence of the complexity of the problem. The most striking observation is the rapid increase in the complexity of the problem as new constraints are imposed. The unconstrained problem of section 4.6.1 can be solved analytically. Once sequencing constraints are introduced (section 4.6.2), the problem becomes exponential. The advantage gained in the dynamic programming approach by classifying aircraft into types is quickly lost when a second runway is introduced in section 4.6.3. Finally all approaches fail when scheduling constraints or departures are introduced.

The general runway scheduling problem (section 4.3) is far more complex than any of the variations that have been examined. Furthermore, a new solution has to be found every time a new aircraft (landing or departing) enters the terminal area ATM/C system. In view

of the restrictions imposed on the algorithm by the environment in which it has to operate, it is necessary to abandon the search for strictly optimal schedules. The alternative is to use heuristic algorithms which will generate near optimal schedules within the time limits imposed by the real time operation of the ATM/C system.

Heuristic algorithms consist of a set of rules that are used to generate new feasible solutions to the problem at hand in a systematic way. The rules are usually local in nature. They are applied to the solution that has thus far resulted in the best value of the objective function, to generate a new feasible solution. Each new solution is, therefore, a local variation of the current best solution. The newly generated solution is compared to the current best and the one producing the best value for the objective function is kept. The algorithm terminates when the current solution is better than all its local variations generated by the algorithm.

There are two disadvantages associated with heuristic algorithms. First, the solution found by the heuristic may be far from optimal. Second, since the heuristic rules are usually local in nature, the value of the final solution may vary greatly depending on the initial solution used to start the algorithm. In many situations, the worst case performance of a heuristic algorithm can be ascertained. An algorithm may, for example, have a "worst case performance of 2". This means that

the value of the objective function resulting from the heuristic solution will be at most twice that of the optimal value.¹

The average loss in performance with respect to the strictly optimal solution is usually much less severe than indicated by the worst case performance of the algorithm. Worst case performance is usually based on pathological cases which are seldom, if ever, encountered in real applications.

A much more indicative measure for a heuristic is the average performance, i.e. how close to the optimal are the heuristic solutions on the average. A heuristic, for example with worst case performance of 2 may, on the average, generate solutions that are within 10% of the optimal. An extreme example of this discrepancy between the average and worst case performance of algorithms is the Simplex method used to solve Linear Programs.² In the worst case, the time required to obtain the optimal solution using the Simplex algorithm is an exponential function of the number of constraints.³ The average performance of the algorithm, however, is a polynomial function of this number. The success of the Simplex method is, understandably, due to this tremendous difference in average versus worst case performance. Unfortunately, the average

(1) Assuming the objective function is to be minimized. The analogous definition applies for maximization of the objective function.

(2) In this case the performance is in terms of time required to obtain the solution since the Simplex method is an exact algorithm. The principle, however, remains the same.

(3) See [PAP 82]

performance of an algorithm is often much more difficult to derive (or even estimate) than its worst case performance.

The heuristic algorithm which has been implemented in TASIM has important advantages which make it a very attractive alternative for this application. We will discuss some of the most important ones.

First, the process of mathematical formulation of constraints and objective function in an optimization problem usually involves a degree of idealization. Linearization of higher order functions is a typical example. In addition, real world applications often impose constraints that are either qualitative in nature, or are not amenable to the mathematical formulation adopted. In the runway scheduling problem, there exist a variety of other operational constraints that the ATM/C controller may want to impose on the schedules. For example, it may be necessary to force an aircraft to land as soon as possible, or to force two aircraft to operate in direct succession. Even if the general problem formulated in section 4.3 could be solved in polynomial time, it would probably be impossible to incorporate the full repertoire of other constraints which would be required to insure that the ATM/C controller remains in control of the automated decision support system. The heuristic procedure can handle arbitrary forms of constraints as well as arbitrary objective functions.

Second, at all times prior to the termination of the heuristic there exists a feasible solution. This is not the case with all optimal

algorithms. In an important subset of them, the dual algorithms, the first feasible solution obtained is the optimal solution. The availability of a feasible solution at all times may be very important from an operational aspect, since within the overall system in which the algorithm is embedded, a number of events may require the optimization process to terminate abruptly.¹ These events are usually not evenly spaced but occur at stochastic intervals. This property can be effectively used in controlling the time allowed for reaching good solutions. When there is plenty of time available the algorithm is allowed to generate better solutions. When the time available is scarce, strict time limits may be imposed on the algorithm and we are still assured of obtaining a good solution.

Third, in real world applications, there is seldom a single objective that should be optimized. This is clearly the case with the problem at hand. Runway capacity, aircraft delays, user costs, etc., are all possible candidates for optimization. Exact algorithms often fail to capture the multiplicity of attributes that constitute a "good" solution. The problem is compounded by the fact that, in the overwhelming majority of real world problems, a substantial number of near-optimal solutions exist. Each may result in a value of the primary objective function that is indistinguishable, for practical purposes, from that of the strictly optimal value. Yet, they may rate

(1) In the case of runway scheduling such an event may be a new aircraft entering the system.

substantially higher than the latter with respect to the secondary objectives. The heuristic is designed to make intelligent trade-offs among alternative solutions based on secondary objectives.

In many situations, including the problem at hand, a heuristic algorithm may be the only viable alternative. The advantages of heuristic procedures may easily compensate for the possibility of adopting inferior solutions, even when the problem, in its idealized mathematical formulation, can be solved through efficient polynomial algorithms.

4.8 A Heuristic Algorithm for the Scheduling Problem

The heuristic algorithm implemented in TASIM uses the basic idea adopted by Dear to generate feasible sequences in a systematic way, [DEA 76]. A number of important improvements have been introduced, however. The scheduling logic has been extended to provide for multiple runways; time varying scheduling constraints have been incorporated; finally, the algorithm has been adapted to the real-time environment in which it has to operate.

The heuristic consists of three parts or stages. In the initialization stage, the problem parameters are set up according to the "state" of the ATM/C system at current time. The second stage is a feasibility search which generates a feasible solution to be used as the initial solution required by the third stage. The latter is the

optimization heuristic which generates new sequences by local permutations of the current best sequence. The algorithm stops when none of the local permutations allowed by the heuristic rule is better than the current best.

4.8.1 The Initialization Stage

The initialization stage of the scheduling algorithm is depicted in figure 4-1. Upon being invoked the algorithm enters this stage. First, the decision aircraft set, A^d , and the set of all aircraft ineligible for rescheduling, A^o , are generated. Aircraft may not be eligible for rescheduling because their current STAR is within a prespecified lead time from the current time. In addition, the ATM/C controller may explicitly "freeze" the STAR for some aircraft. Finally the flight plan generation algorithm may constrain a landing from being rescheduled if its position and its surrounding traffic pattern do not allow modifications to its flight plan to be generated.

The scheduling constraints for all aircraft in the system are updated next. The updating procedure is related to the position of the aircraft along its previously assigned flight plan and the runway assigned to it. This will be discussed further in connection with automatic flight plan generation which is presented in the next chapter.

The time separations between all pairs of aircraft are calculated next. These are used during the feasibility and the optimization

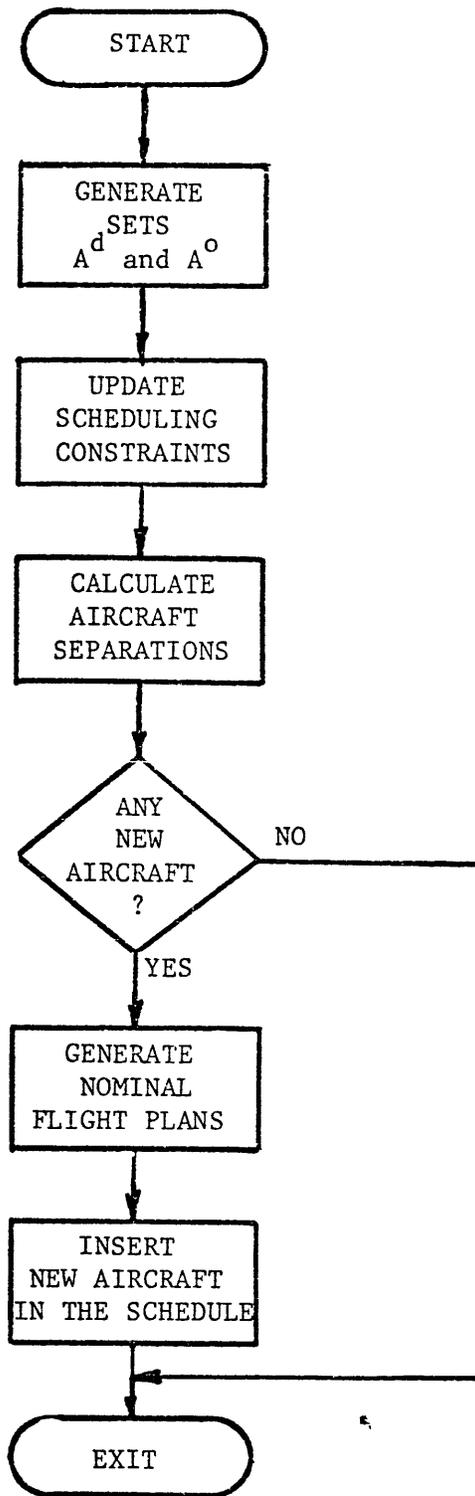


Figure 4-1. Runway Scheduling Heuristic, Initialization Stage.

stages. Special separation requirements that the ATM/C controller may want to impose on specific aircraft are taken into account during this calculation.

The most common reason for invoking the scheduling algorithm would be the entry of a new aircraft in the system. New aircraft are incorporated in the optimization process by first generating nominal flight plans for them, and by inserting them in the current schedule. The nominal flight plan generation is based on prespecified nominal landing and departing routes as well as nominal speeds along these routes. Nominal flight plan generation will be discussed further in the next chapter.

Landing aircraft are initially inserted in their nominal position in the sequence. In most cases, the nominal position for landings will be at the end of the sequence and the resulting schedule will be feasible. It is possible, however, that the aircraft's nominal position is not at the end of the current schedule. This situation will typically exist when travel times from various entry fixes to the runway thresholds differ substantially. In this case, it is possible that the schedule will be infeasible. Whenever the schedule is not feasible, the aircraft is moved backward in the sequence until a feasible schedule results or until it reaches an infeasible position. Of course, when two or more runways are active, the nominal as well as subsequent positions on each active runway are tried before moving the aircraft backward.

Note that the relative sequence of all other aircraft is not changed by the insertion.

New departures are treated somewhat differently because the difference between the current time and their preferred time of arrival at the runway is based on taxi time, which is typically much shorter than the time required for landings to reach the runway from an entry fix. An attempt is made to insert new departures in the sequence as far forward as is allowed by their PTAR. If a gap between two landings is large enough to allow the departure to take off without changing the existing scheduled time of either landing, the departure will initially be assigned to this slot. The assignment may of course be changed during the optimization stage. Again when two or more runways are active all possibilities are tried before the departure is moved backward in the sequence.

Finally, once all the new aircraft have been incorporated in the schedule the constraint list is updated to include any new constraints that have been imposed by the ATM/C controller or the flight plan generator. During this process obsolete constraints (e.g. ones having to do with aircraft which are not in the decision set A^d) are deleted.

4.8.2 The Feasibility Stage

In general, the schedule generated by the initialization stage of the algorithm will not be feasible. The algorithm will, therefore,

enter the feasibility stage. A feasible schedule is generated by using the same heuristic procedure as the one employed in the optimization stage to generate the optimal schedule. The only difference is that, while the normal objective function(s) are used to compare schedules during the optimization stage, the feasibility stage uses a specially defined function called the index of infeasibility. This function is defined so that its value is zero when the schedule is feasible, and positive if the schedule is infeasible. In essence, the method used to generate a feasible schedule is akin to the standard method employed to generate a feasible solution to a linear program.¹

A variety of such functions can be defined. The most straightforward method is to define the objective as the sum of all the infeasibilities. By infeasibility, here we mean the amount by which each constraint is violated. Of course, the constraints that are not violated do not contribute anything to this sum. As an example, suppose that according to a schedule

$$STAR_{in} > LFTAR_{in}$$

for some aircraft i and some runway n . The schedule is obviously infeasible and the contribution of the scheduling constraint for aircraft i to the infeasibility index would be equal to $STAR_{in} - LFTAR_{in}$.

(1) See for example [SIM 66].

During the feasibility stage therefore, the current best schedule is the one for which the index of infeasibility is least. If at any point the value of the infeasibility index goes to zero the corresponding schedule is feasible and the algorithm enters the optimization stage with that schedule as the initial solution.

4.8.3 The Optimization Stage

The optimization stage is based on the algorithm constructed in section 4.5. The primary motivation is that, given the values of the 0-1 variables of the formulation in section 4.3, the optimal schedule can be easily constructed based on a scheduling assumption consistent with the objectives.

In practice, the optimization is performed in three sequential steps: sequence generation, schedule generation, and the schedule evaluation. The flowchart of figure 4-2 depicts the combined operation of the feasibility and the optimization stages.

The method for generating sequences of operations to be evaluated is independent of the number of runways that are active. The input (current best) schedule is used to define the base sequence of operations according to their STAR's, as if all aircraft were operating on a single runway. At any time, the positions of aircraft within a small subsequence are permuted to produce the new sequence. We will

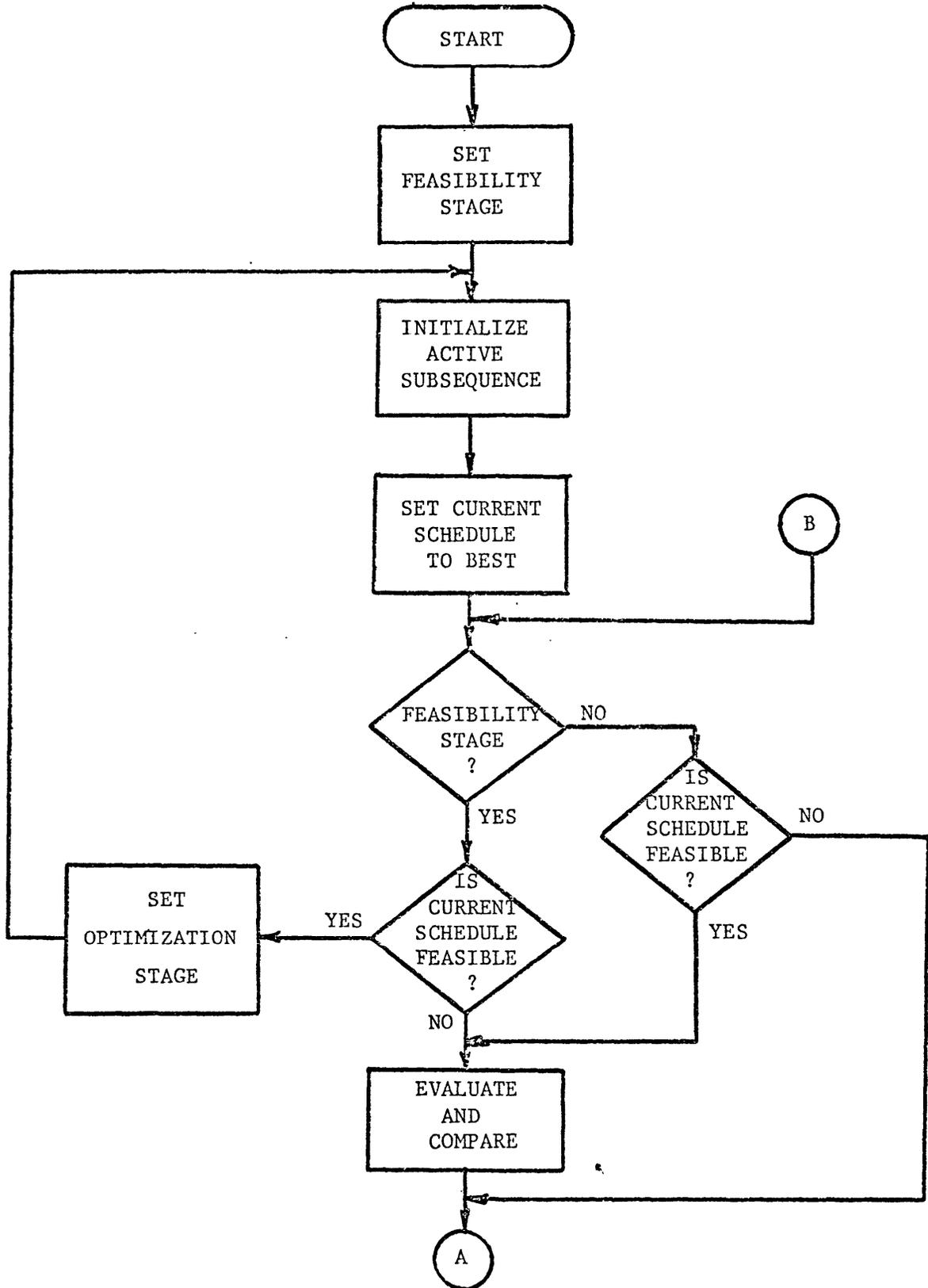


Figure 4-2. Runway Scheduling Heuristic, Feasibility and Optimization Stages (continues).

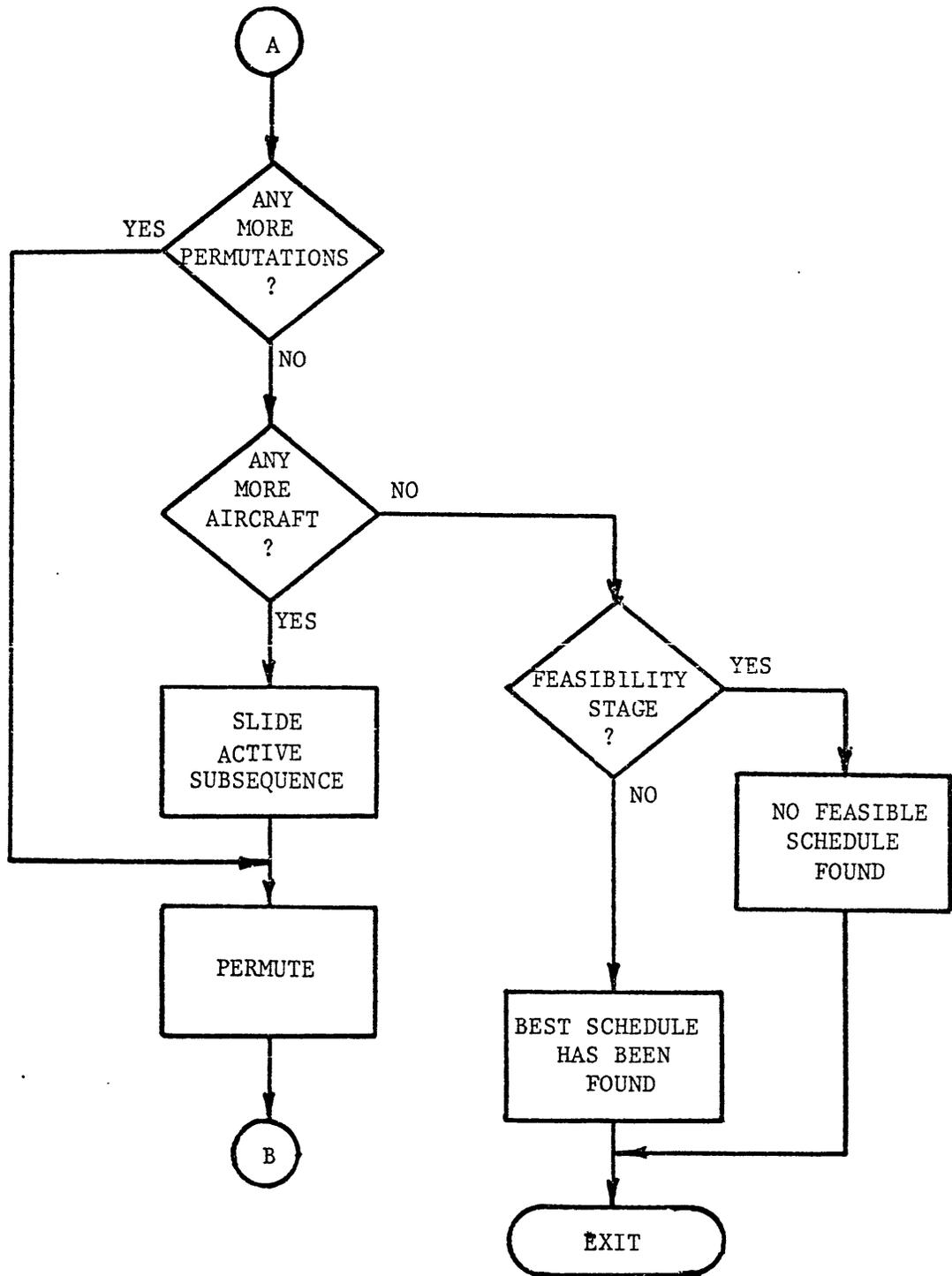


Figure 4-2. Runway Scheduling Heuristic, Feasibility and Optimization Stages (continued).

call this the active subsequence. The positions of aircraft which are not within the active subsequence are not affected.

The procedure begins by defining the active subsequence to include the last K aircraft in the base sequence, where K is an input parameter. The next permutation within the current active subsequence is generated and checked for feasibility with respect to the sequencing constraints relative to the overall sequence (constraints 3c and 3d of section 4.3). If all the permutations within the current active subsequence have been evaluated, the subsequence slides forward. The position of the last aircraft in the subsequence becomes permanent and the aircraft immediately preceding the first aircraft of the subsequence is included in its place.

Once a new sequence has thus been generated, the corresponding schedule and runway assignment can also be determined based on the scheduling assumption. This is done by sequentially considering each aircraft, starting with the first aircraft in the current permutation of the active subsequence and ending with the last aircraft in the overall sequence. Note that only the schedule of aircraft succeeding the ones that have been resequenced could be affected.

If the runway assignment of the aircraft under consideration has not yet become permanent,¹ the scheduled times of arrival at the runway

(1) Runway assignments usually will become permanent before the scheduled time of arrival at the runway is frozen.

are evaluated for all possible runway assignments and the earliest STAR is used. A runway assignment may be infeasible due to the sequencing requirements within the runway, (constraints 2a and 2b of section 4.3). In addition, a runway assignment may be infeasible for operational reasons. For example, the runway may be too short for the type of aircraft in question, or the controller may have explicitly requested that aircraft of that type should not use a certain runway.

Given the runway assignment, the scheduled time of arrival at the runway is determined so that it satisfies the spacing constraints with respect to all aircraft for which STAR's have already been determined. In addition, it has to comply with the earliest feasible time of arrival at the runway $EFTAR_{in}$. $STAR_{in}$ is therefore given by:

$$STAR_{in} = \max \{EFTAR_{in}, t_{in}\}$$

where,

$$t_{in} = \max \{STAR_{jRWY(j)} + s_{jiRWY(j)}\}$$

The maximization determining t_{in} is performed over all aircraft j preceding i in the sequence under consideration. If at any point during the schedule generation, any of the remaining constraints is found to be violated, the process is stopped and a new sequence is generated.

The evaluation of each schedule generated captures the multi-objective nature of the runway scheduling problem. A number of alternative objective functions are "active" simultaneously. At any time, the ATM/C controller can decide what the relative importance of

each active objective should be. Thus, minimization of the average aircraft delay may be the primary objective, maximization of the capacity of the runway system may be the secondary objective, and minimization of the absolute number of position shifts may be the tertiary objective. The decision will presumably be based on the current traffic situation in the terminal area.

The value of all active objectives is determined. The comparison is first done based on the primary objective. The secondary objective is used to break ties between schedule that minimize the primary objective. If ties still exist, the tertiary objective is used, and so on until all the active objectives have been scanned. In the unlikely event that ties still exist, the current best schedule is preserved.

Two schedules are considered equivalent with respect to a certain objective if their values are within a certain percentage of each other. This allows, for example, two schedules which result in average delays of 2.7 and 2.8 minutes respectively, to be considered equivalent and thus their performance with respect to other objectives becomes the deciding factor. The actual "margin of equivalence" within which the schedules are considered equivalent can vary depending of the objective in question.

The new schedule replaces the current best if it is found to be better. When this happens, the sequence generation process is restarted with the last K aircraft comprising the active subsequence once again.

Accordingly, the optimization stage terminates when the forward sliding of the active subsequence will include an aircraft which is not in the decision aircraft set.

4.9 Implementation Status

The heuristic algorithm described in the previous section has been implemented and is currently operational in the real time environment provided by TASIM. The software can handle both landings and takeoffs, as well as multiple runway configurations. Some of the features which were described in the previous section for the sake of completeness have not been implemented, however, because they require the existence of software for flight plan generation, as well as full specification of the controller interaction with the runway scheduling software. These are:

1. Software for generating nominal flight plans and for determining the scheduling constraints applicable to each aircraft. This software will be implemented fully in conjunction with the Traffic Flight Plan Generator. Currently nominal flight plans are obtained as input data and are specific to each entry fix.
2. Software to accept and process controller imposed constraints. The heuristic is designed to allow a wide variety of constraints. The full repertoire of constraints

which should be available to the controller, however, has not been specified.

3. Software for determining the separation minima. Currently the separation minima are obtained as input data. This is adequate for the purposes of research and development. In a fully operational system, however, separations should be determined based on the actual traffic. This will eliminate classification of aircraft into prespecified categories and will allow the runway scheduling algorithm to treat each aircraft as unique.

There has been limited simulation testing of the runway scheduling heuristic. It appears to provide a very practical method for obtaining an efficient runway schedule at very high operational rates. Further testing is needed, however, in order to establish its performance and demonstrate the efficiency gains in terms of increased runway operational capacity, and reduced delays under given traffic conditions.

4.10 Coordination Among Terminal Area Control Sectors

Automatic runway scheduling represents a drastic departure from current ATM/C procedures in the terminal area. In particular, implementation of this function will require restructuring of the

coordination procedures among the terminal area control sectors (i.e. approach control, departure control, ground control,¹ and Tower).

In the present system, the Tower is responsible for takeoff and landing clearances, but does not have direct control over the landing stream it receives from final approach control, or over the departure stream it receives from ground control. This means that Final Approach is responsible for sequencing landings, Ground control is responsible for sequencing takeoffs, and the Tower is responsible for interlacing the two types of operations.

With the implementation of runway scheduling, all three of the above decisions are centralized. Assuming for the moment that the current controller positions in the terminal area remain distinct, it is clear that substantial coordination will be required among them. At this point we cannot provide an answer as to how this should be done. We will, however, briefly describe one possible scenario:

The Tower controller has primary responsibility for all major decisions regarding the use of the runway system. With the support of the software, he generates the runway schedule. He also maintains his current responsibilities, i.e. assuring the safe operation of the runway system.

(1) Here ground control is the terminal area sector responsible for controlling traffic on the surface of the airport and not about the ground control system in general.

The Tower controller normally interacts with the runway scheduling software for the purpose of establishing the general strategy. He decides, for example, which objective function should be optimized, what MPS value is appropriate for the particular time of day, etc. He may, however, impose tactical constraints (e.g. specifying the position and/or the STAR for some aircraft, requiring that two aircraft exchange position in the sequence, forcing the software to schedule some landing at the earliest possible time, etc.) if the need arises.

The Tower controller also makes decisions regarding the runway configuration to be used and informs the scheduling function of unusual situations that have to be taken into account. For example, he enters a missed approach for rescheduling, etc. Finally, he reviews constraints that other controllers may have imposed and negotiates changes if required.

The current schedule for departing aircraft is made known to the Ground controller and to the Departure controller. The Ground controller is responsible for the management of the departure queue. He has to insure that the order in which aircraft are queued at each runway holding area is the one required by the schedule. Since a precise schedule is available, the ground controller does not have to implement gate hold based on the length of the departure queue. Gate hold decisions can now be made on an aircraft-by-aircraft basis. The gate hold interval will depend on the scheduled delay and on the estimated taxi time for each individual aircraft.

The Final Approach controller obtains scheduling information on all landing aircraft. He is responsible for delivering landings at the Outer Marker at the time implied by the schedule. He interacts with the schedule by imposing constraints whenever, in his judgement, the schedule cannot be safely implemented. He is, therefore, the one most likely to use the tactical constraints which were mentioned above in the discussion of the responsibilities of the Tower controller.

Finally, the Entry controller is responsible for landing traffic as it first enters the terminal area. He, also, has complete information on the schedule for landing aircraft and manages the operation of holding stacks (when they are needed) in the same way the Ground controller manages gate hold operations. It is important to note again that runway scheduling allows holding stack management to be done on an aircraft-by-aircraft basis.

4.11 Computer Aided Vectoring for Approach Spacing

It is impossible to expect that a fully automated ATM/C system can be introduced at once. For this reason, it is important to consider the evolutionary period when automation functions are gradually in order to determine the interim needs of air traffic controllers.

In order to exemplify how TASIM can be useful in addressing this type of problem, a special purpose display was designed and implemented.

The display is designed to assist the final spacing controller in achieving accurate delivery of landing aircraft at the runway without the support of automated flight planning. The display provides the controller with a visual method to time the final turn onto the runway centerline.

The geometry of the final approach area, shown in figure 4-3, is part of the final vectoring display (also figure 3-8). The final vectoring area (shaded in figure 4-3) starts at a point called the vector marker (VM) which is situated about 2 miles from the outer marker (OM). The OM is the point where the aircraft would acquire the glideslope. It is assumed that the final vector will be less than $\pm 20^\circ$ to the runway centerline so that the pilot/autopilot may acquire and commence tracking the centerline with less than a 20° change in heading. The displacement of the VM is designed to allow stabilization on the runway centerline by the time the OM is reached, even if the aircraft is vectored along the edge of the final vectoring area and thus intercepts the centerline at the VM.

Horizontally across the top of the display, there are boxes representing arrivals from the left or the right side of the runway. The number inside each box indicates the position in the landing sequence currently assigned to the aircraft by the runway schedule. The boxes move towards the runway centerline extension at a speed equal to the landing speed of the aircraft they represent.

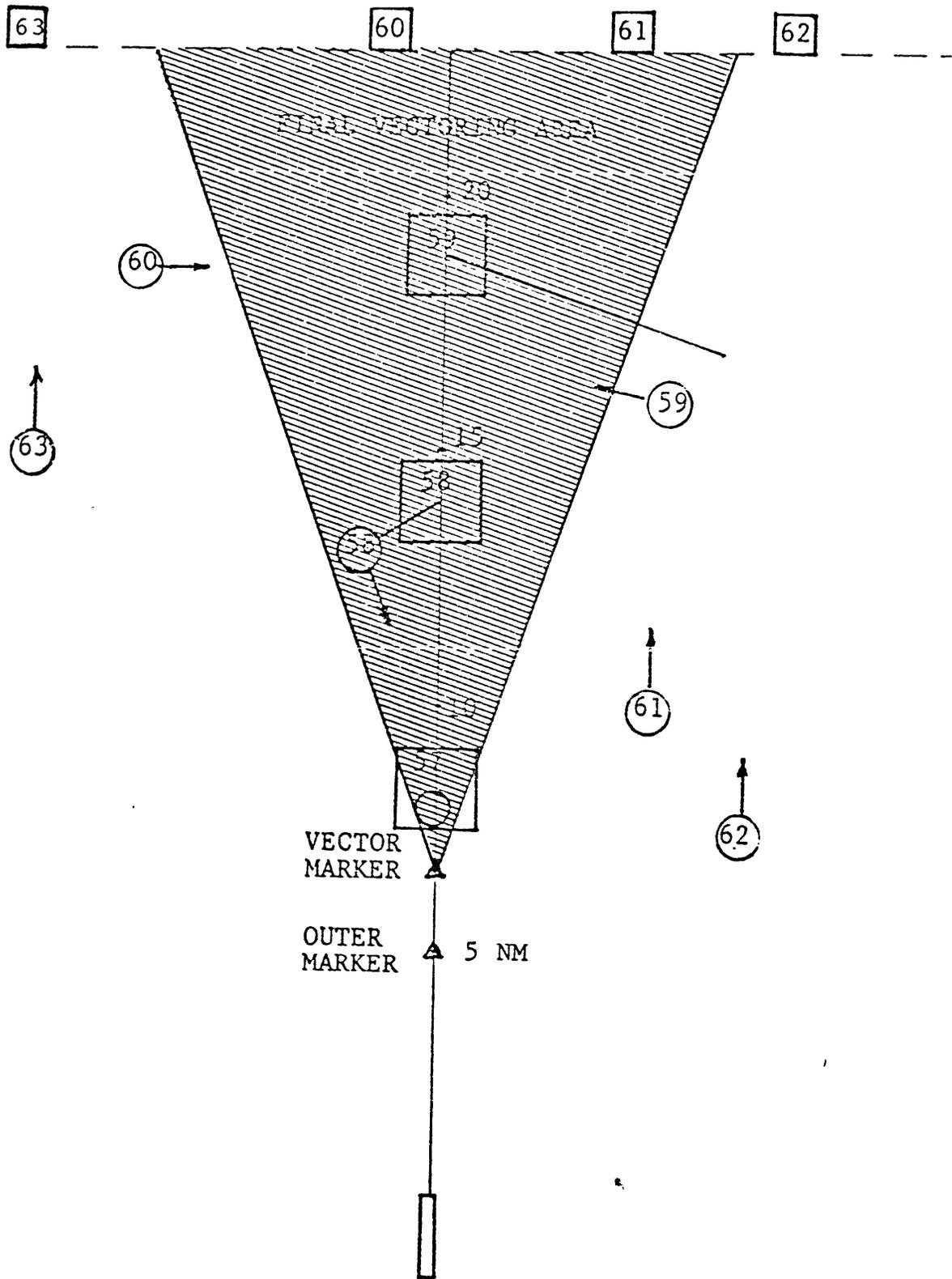


Figure 4-3. Final Vectoring Display.

When a box reaches the runway centerline extension,

1. it expands to a size which indicates the maximum tolerable error in the delivery to the runway. This value has been specified, and will generally depend on the accuracy of the surveillance system.
2. it grows a "wand" to the left or right (depending on the position of the aircraft it represents), and
3. it starts moving along the centerline extension towards the OM.

The wand is positioned so that when it touches the target, it is time to call the final vector to intercept the runway centerline at a 20° angle. If this procedure is performed properly, the target will intercept the box on the centerline at some point before the VM.

This procedure has currently been mechanized for the final turn only. It can be easily extended, however, to help the controller with the timing of the turn to the base leg as well. Finally, it can also provide the timing for calling the deceleration to the final approach speed. Appendix B examines how the positioning of the wand can be determined as a function of the aircraft current and landing speeds.

CHAPTER 5

A METHODOLOGY FOR TRAFFIC FLIGHT PLAN GENERATION

5.1 Introduction

Optimal scheduling of runway operations provides the potential for substantial improvements in the efficiency of the runway system. This potential can only be realized if the schedule is achieved. This means that landing aircraft must be delivered at the threshold of their assigned runway on time.¹

To accomplish precise delivery at the runway, 4-dimensional, conflict-free flight plans are generated for all landing aircraft. These are based on the aircraft's current position, as determined by the surveillance and tracking system, and the aircraft's STAR, as determined by the runway schedule.

Flight planning, performed by the Traffic Flight Plan Generator (TFPG), establishes with certainty that the runway schedule is feasible, i.e. there exists a safe (conflict-free) set of paths which will satisfy the schedule.

Flight planning is only the first step towards accomplishing precise delivery of landings at the runway.² Once flight plans are

(1) Obviously departures should also be on the runway ready for takeoff at their scheduled time. This, however, does not involve flight planning.

specified, there remains the problem of guiding the aircraft along their assigned path. Aircraft guidance is accomplished through the interaction of the Command Generator, which determines a set of commands suitable for each aircraft's navigation capabilities, the Command Activator, which activates and dispatches these commands in a timely fashion, and the Conformance Detection and Conformance Resolution functions, which insure that the flight plans are indeed being followed. Due to the close relationship between flight planning and aircraft guidance, the methodology presented here will, on many occasions, encompass the guidance functions as well.

Consistent with the basic design philosophy of the automated terminal area ATM/C system, it is necessary to maintain the master-slave relationship between the controller and the TFGP software. This implies that flight plans should be easily understood and visualized by the ATM/C controller. Accordingly, the horizontal profile (ground track) of the flight plans will be composed of a small number of linear segments or legs. The third and fourth dimension of the flight plan will be provided by specification of altitude and speed change points along some of the legs.

Advanced navigation and flight control systems onboard the aircraft will enhance the operation of the ATM/C system. They should not, however, be a prerequisite for using it. Piecewise linear horizontal profiles are also consistent with this design goal for the TFGP and the automated terminal ATM/C system in general.

The TFPG has to generate flight plans that adapt to a dynamically changing runway schedule. To achieve this, ground tracks are selected from a prespecified path structure which allows a number of alternative paths to be generated for an aircraft at any point along its approach. As a result, considerable flexibility in path stretching and path shortening exist until the late phases of the approach, when the aircraft's scheduled time of arrival at the runway becomes fixed.

The changing runway schedule has important implications with regard to aircraft guidance as well. During the early part of the approach of any aircraft, when its STAR will be changing, there is no need to strictly enforce the 4-dimensional flight plan. The Conformance Detection function is still used to identify non-conforming aircraft. However, instead of forcing the aircraft back into conformance, the TFPG uses estimates of their current position, ground track, and ground speed to generate new flight plans. In other words, in the early phases of the approach we can treat non-conformance as if the runway schedule had changed. Thus, the inherent flexibility of the flight path selection process is used to correct navigation errors, errors in the estimates of the prevailing winds, and finally, as a means of avoiding active speed control on most occasions.

Changes in the runway schedule and conformance alerts are the most usual events that will result in modification of an aircraft's flight plan. In some cases, however, a revised flight plan will have to be generated for some aircraft in order to resolve conflicts which have

resulted from modifications of flight plans of other traffic. Finally, the ATM/C controller may impose changes in the flight plan, either by issuing a direct command to the pilot, or by specifying a new flight plan for the aircraft to the TFPG function.

The implication of the above scenario is that, flight planning for the terminal area ATM/C system is accomplished by repeatedly solving the same basic flight plan generation problem stated in chapter 2 (section 2.3.3). At each point in time when new flight plans are required, the origin (i.e. the current position) and the destination (i.e. the runway threshold) for each landing aircraft in the terminal area is precisely specified in 4 dimensions. The flight planning task is to find a set of 4-dimensional, conflict-free paths which satisfy the above boundary conditions.

The link between consecutive invocations of the traffic flight plan generation function is the solution itself. Namely, the aircraft positions (i.e. the input to the TFPG) are a direct consequence of the flight plans (i.e. the output) which were generated at some earlier point in time. Even though this relationship is obvious, it has to be pointed out. It emphasizes that flight plans must leave open as many options as possible in order to allow modifications, if the need for them arises at some future time. This flexibility is particularly needed in the initial phases of the approach.

5.2 Flight Path Structures

A typical layout of the Terminal Control Area (TCA) is shown in figure 5-1. The TCA extends 30 to 60 nautical miles from the airport, which is situated in the center of the figure. A number of waypoints at the periphery of the TCA have been designated as entry and exit fixes. Landing traffic is directed by enroute controllers towards the entry fixes, (shown as upward pointing triangles) along jet routes or airways. Departing traffic is directed towards the exit fixes, which are shown as downward pointing triangles. The exit fix assignment is determined based on the destination airport of the departure. Landings are handed off to the ATM/C controller 5 to 10 nautical miles prior to reaching the entry fix. Departures are handed off to enroute sectors 5 to 10 nautical miles prior to reaching the exit fix. Nominal approach routes from the entry fixes to the runway are shown in figure 5-1 in solid lines. The dashed lines represent nominal departure routes from the runway towards the exit fixes.

The nominal approach route from entry fix A to the runway is shown in greater detail in figure 5-2. It consists of 5 linear segments or legs.

1. The Entry leg, (OA in figure 5-2)
2. The Initial Approach leg, (AB in figure 5-2)
3. The Downwind leg, (BC in figure 5-2)
4. The Base leg, (CD in figure 5-2)
5. The Final Approach leg, (DE in figure 5-2)

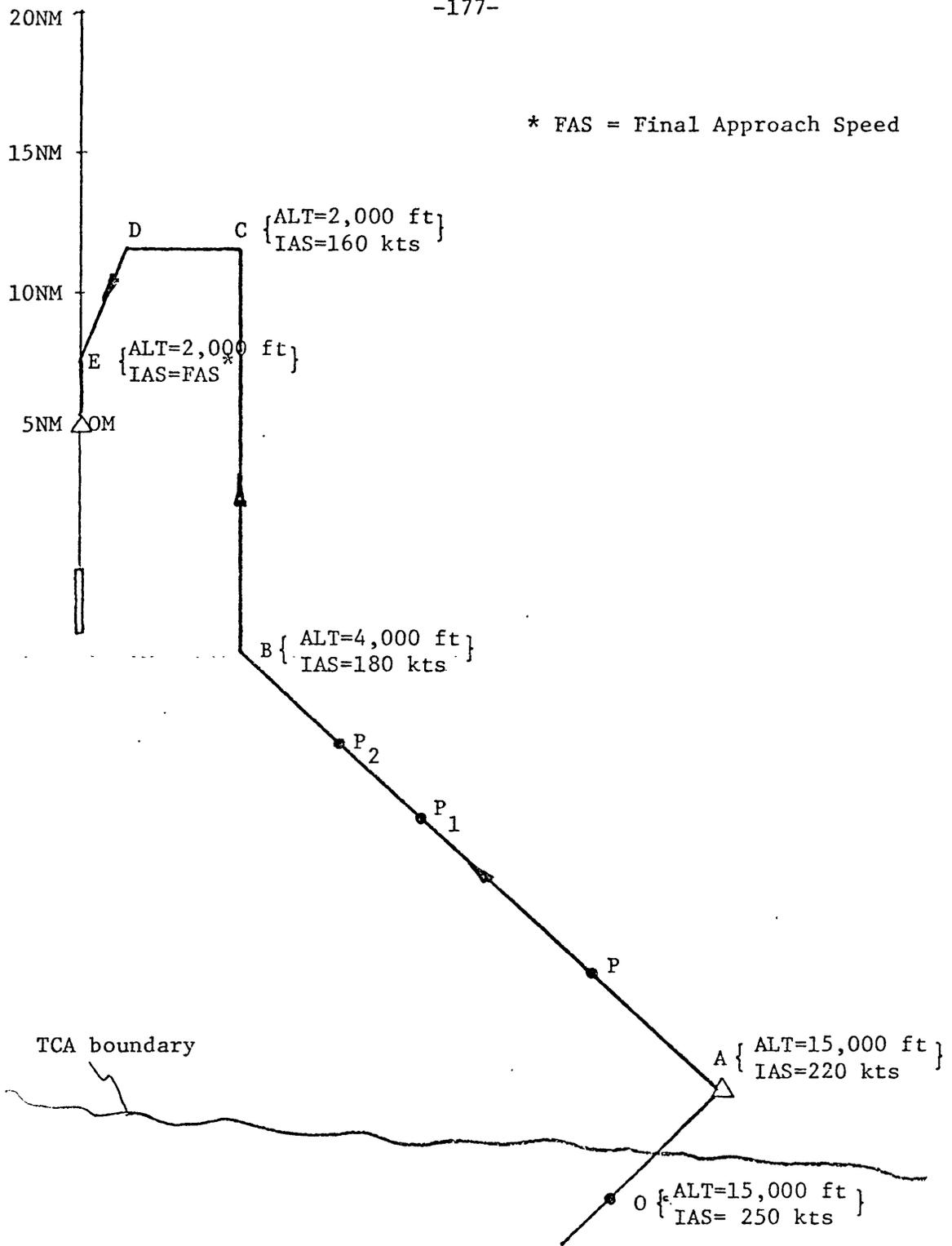


Figure 5-2. Typical Nominal Approach Flight Plan.

Note that the leg from point E to the outer marker, (OM), is also considered to be part of the final approach leg. Furthermore, we have assumed that an outer marker will continue to exist until the ILS is decommissioned.

Each leg is associated with a particular region of the terminal area airspace, and with the same general direction of traffic. The location of the entry and downwind regions, as well as the direction of traffic in them, depends on the location of the entry fix. The downwind, base, and final approach regions, on the other hand, are specific to each runway direction. This means that the downwind and base regions are merging points for traffic from two or more entry fixes.

From nominal approach routes, we can construct nominal flight plans by specifying altitude and speed profiles along each leg. Nominal flight plans are based on typical values for the airspeed and descent rates at the preferred aircraft configuration (e.g. airspeeds resulting from "idle thrust" descents at various flap settings). Consequently, they may vary depending on the aircraft type. Figure 5-2 shows typical altitudes and indicated airspeeds, (IAS), at various points along the nominal approach route for jet aircraft.

Nominal flight plans can be thought of as the ideal 4-dimensional flight profile from the entry fix to the runway for each type of landing aircraft. Thus, they determine the preferred time of arrival at various

intermediate waypoints as well as PTAR, the preferred time of arrival at the runway for each landing aircraft. The latter, as we have already noted, is used to implement the sequencing constraints for the runway scheduling function.

The TFPG associates each leg of the nominal route with a distinct phase of the approach. The phase in which the aircraft is when its flight plan is revised, determines the flight planning options that are available. An aircraft which is in the initial approach phase (e.g. at point P in figure 5-2) may transition to the downwind phase at different distances from the runway centerline, (e.g. at points P_1 or P_2), and fly along a downwind leg which is parallel to the one shown in the figure. Similarly, a choice of base legs is available to that aircraft. As soon as the aircraft transitions to the downwind leg, a significant part of its flexibility is lost since only the base leg selection remains available.

The geometry of the ground tracks has important effects in the performance of the TFPG, particularly with regard to landing aircraft, which have to achieve a precise and tight schedule at the landing runway. Many of the characteristics we will discuss, however, are equally as important in flight planning for departing traffic.

First, in order to avoid the accumulation of errors as the aircraft proceeds along its flight plan, small errors in the delivery of the aircraft at intermediate waypoints must be readily absorbed by

modifications during subsequent legs of the ground track. During the initial approach and the downwind phases, this capability will substantially limit the need for strict speed control. During the later phases of the approach, it will determine the accuracy with which landings can be delivered to the runway.

Second, a multiplicity of ground tracks capable of delivering aircraft to the runway at the scheduled time should be available. This will insure with high probability that a set of conflict-free flight plans exist.

Third, scheduling flexibility should be maintained at every point along the flight plan to the highest degree possible. The interval between EFTAR and LFTAR will generally decrease as the aircraft proceeds along the various phases of the approach. This does not present any difficulties since rescheduling during the late phases is not desirable anyway. Some flexibility needs to be maintained, however, even during the base phase. It will be necessary in order to allow small schedule changes brought about by failure of preceding operations to meet their schedule. In many respects, this characteristic is equivalent to the capability to absorb delivery errors at intermediate waypoints.

In order to obtain these characteristics, we specify path structures which provide a number of alternative legs for each phase of the approach. There are many alternative structures which will accomplish this. To be concrete, one such possibility will be described

(figure 5-3). The solid line depicts the nominal approach route which is identical to the one shown previously in figure 5-2.

Even though each phase of the approach requires somewhat separate treatment, we can distinguish two major stages during which the goals in generating flight plans differ greatly. The first stage, which we call adaptive, starts when the ATM/C system first obtains information on an incoming flight and includes the entry, initial approach and downwind phases. The second, or precision, stage includes the base and the final approach phases. Each aircraft transitions from the adaptive to the precision stage when its scheduled time of arrival at the runway becomes fixed.

The dashed lines in figure 5-3 represent the alternative legs that are available during the adaptive stage. Thus, for the initial approach phase, the aircraft can be routed along leg AB or along any leg parallel to AB. For example, the ground track AA_1B_1 is an acceptable alternative for this phase of the approach. Legs parallel to BC provide alternatives for the downwind phase.

During this stage, the aircraft's STAR is changing. The primary flight planning goal, therefore, is to maintain the time interval [EFTAR, LFTAR] as large as possible so that runway scheduling flexibility is maximized. Due to changes in the sequence of runway operations, there will be cases where one aircraft will need to overtake another during this stage. Since all aircraft will be descending while in the

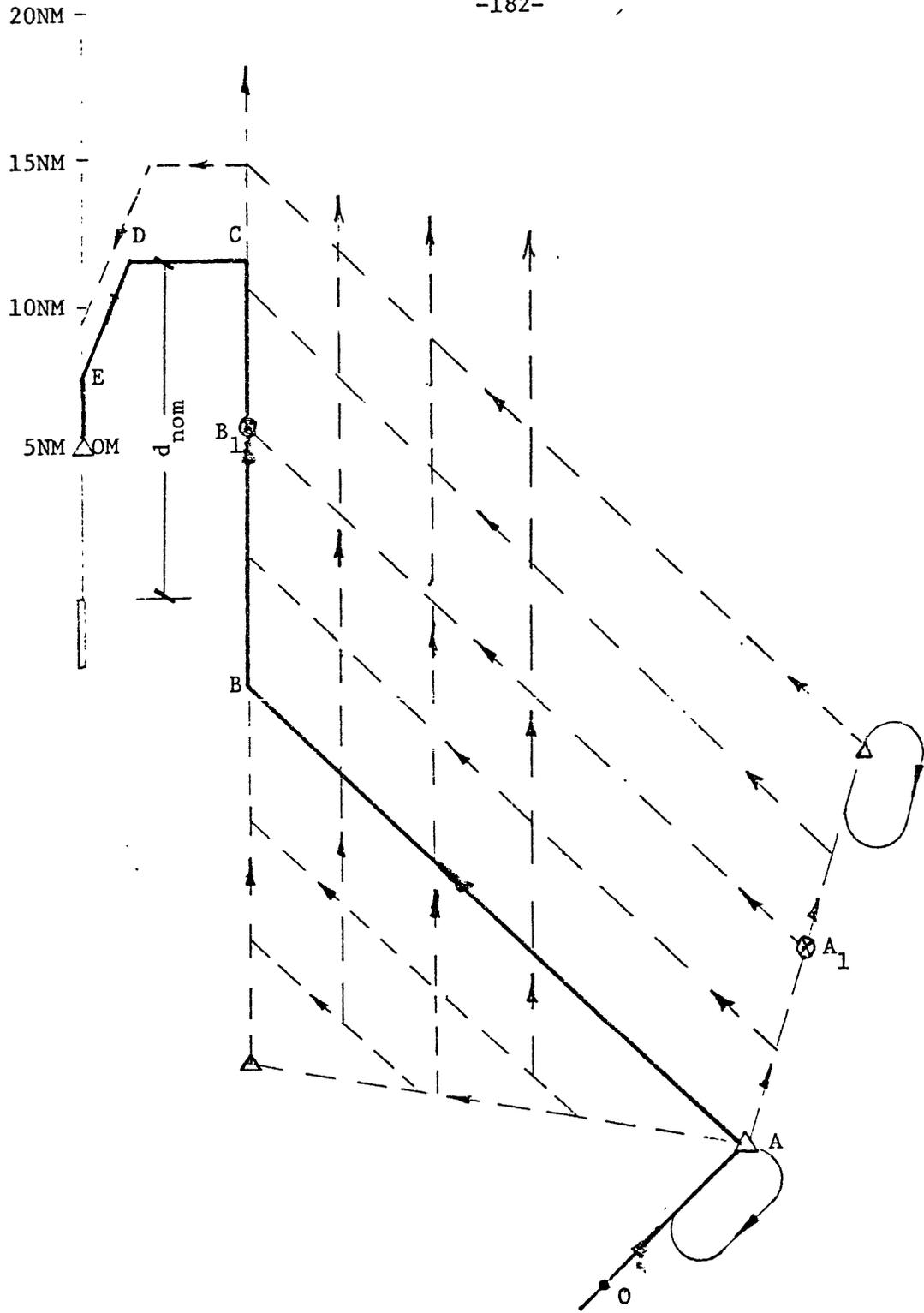


Figure 5-3. Example of an Approach Path Structure.

initial and the downwind phase, it may not be possible to maintain altitude separation between overtaking traffic. Such aircraft will, therefore, need to be on two different legs. In order to insure that there will be no interference between overtaking traffic, adjacent legs in the initial and downwind phase will be spaced 3-5 nautical miles apart.

The selection in the base phase is made from a number of legs which are perpendicular to the runway centerline extension. In the final approach phase, the legs intercept the runway centerline extension at a small angle (typically 20°). During the base and final approach phases, the aircraft's STAR is fixed and flight planning is primarily concerned with precise delivery of the aircraft to the runway. This precision cannot be achieved if ground tracks are restricted to legs that are spaced 3 or 5 nautical miles. Thus, the turns to base and final approach legs do not occur at prespecified points. Instead, the flight plan generator will determine the time and position at which the aircraft should transition from the downwind to the base phase and from that to the final approach phase. One such selection is shown in figure 5-3.

In order to achieve the precision necessary for the proper operation of the automated ATM/C system, the timing of the deceleration to the final approach speed will be controlled during the final approach phase. The final approach will, therefore, be the only phase during

which active speed control will be exercised. At all other times, speed control will be enforced only if no other alternative is available.

The structures used in this particular geometry were chosen because they are widely used for manual spacing in the present terminal area ATM/C system. The structure for the initial approach phase is commonly called a "harp", while the structures for the downwind, base and final approach phases collectively form a "trombone". Analytical models describing the operational characteristics of these and other ATM/C structures have been developed by Simpson, [SIM 64].

5.3 Ground Track Selection and Speed Control

We will present the general methodology for ground track selection and speed control using the selection of the initial approach leg as an example. The method of selection for subsequent approach phases follows the same principles.

Let us consider an aircraft which is in the entry phase of the approach. Referring back to figure 5-3, the aircraft may be at some point O flying towards the entry fix A. We first have to select a specific initial approach leg. This selection cannot be made until we have specified the legs and the speed changes for all the subsequent phases of the approach. The selection problem can, therefore, be viewed from a different perspective. Namely, we can first determine what is a desirable flight plan for the downwind, base and final approach phases.

and then select an initial approach leg based on that flight plan. We will call such flight plans desired flight plans.

In most cases, desired flight plans will be very similar to nominal flight plans, since both pertain to preferred airspeeds, altitude profiles etc. However, we prefer to use distinct terms because: first, nominal flight plans encompass all the phases of the approach, whereas desired flight plans only include a specific subset of phases, and second, nominal flight plans are preferred from the point of view of the aircraft pilot alone, while desired flight plans take into consideration scheduling and flight planning goals as well.

A concrete example will help illustrate the major issues involved in the determination of the desired flight plan for any aircraft. Some additional considerations which pertain to specific landing sequences will be discussed in section 5.6.

For the final approach and the base phases, the desired flight plan may be the same as the nominal. This means that the base leg is at a distance of 12 nautical miles from the runway threshold (d_{nom} in figure 5-3), the turn to final approach is such that the intercept point E is 8 nautical miles from the runway threshold, and the deceleration to the final approach airspeed occurs immediately following the acquisition of the runway centerline extension. The rationale for these selections is that the control points are approximately centered within their respective selection range. It is assumed, therefore, that the base leg

can be at distances ranging between 6 and 15 nautical miles from the runway threshold, and that the turn to final approach can be such that the intercept point E can range from very near the OM to very near the point of intersection of the base leg and the runway centerline extension.

Similar considerations apply to the selection of the desired flight plan for the downwind phase. For example, the speed profile for the desired flight plan can be identical to the nominal speed profile. In determining the desired downwind leg, however, the current position the aircraft occupies in the runway sequence has to be considered also. "Centering" the control point (i.e. the point to turn in this case) would be adequate if the position of the aircraft in the current runway schedule is the same as its nominal position, POS_{nom} (see chapter 4). If, on the other hand, the aircraft has been moved forward or backward from its nominal position, its potential for being moved again by future rescheduling is no longer the same in the two directions. For example, assume an MPS value of 5, and that the aircraft has already been moved backward 3 positions. It can only be further rescheduled backwards by at most two positions. At the same time, it is possible to be rescheduled forward by as many as 8. This implies that the selection of the downwind leg has to be biased accordingly. In the situation of figure 5-3, the downwind legs which are at greater distances from the runway centerline would be preferable to leg BC.

Once the desired flight plan for the downwind, base and final approach phases has been selected, we can determine, for any leg of the initial approach, the point where a speed command has to be given so that the aircraft reaches the intersection with the already chosen downwind leg at the appropriate time. The initial approach leg for which the timing of the speed reduction most closely matches the nominal speed profile will be selected. In the computation of the time for the speed reduction, nominal speeds for both before and after the deceleration can be assumed. For example, typically jet aircraft will decelerate from their entry airspeed of about 220 knots to an airspeed in the vicinity of 180 knots during the initial approach phase. Similarly, during the downwind phase aircraft will lower their flaps which will cause approximately a 20 knot decrease in airspeed.

The flight planning logic applies limits to the timing of the speed change in each phase of the approach. These limits reflect empirical knowledge on when (or at what point) speed changes occur along the approach phase in question. During the initial approach phase of figure 5-3 for example, we may specify that the speed reduction in the flight plan should occur within 10 miles from the entry fix. If the timing of the deceleration, as determined above, occurs within those limits, the desired flight plan together with the initial approach leg and the associated speed change is accepted pending altitude assignment and conflict check.

In some cases, the speed reduction will not fall within the allowable limits. This simply means that the desired flight plan cannot be implemented as is. Since the chosen leg has been the "best" with respect to the desired flight plan, however, it is assigned to the aircraft and the speed reduction is set to occur at the earliest (or latest) allowable. Having thus selected the leg and speed profile for the initial approach phase only, the TFPG will proceed to select an appropriate leg and the corresponding speed profile for the downwind leg. The selection is again based on the final approach and base phase portions of the desired flight plan. This iterative procedure continues until the complete flight plan has been generated.

5.4 Altitude Selection

Aircraft altitudes will be actively controlled to achieve the required separations among aircraft whenever adequate longitudinal separation cannot be provided. Within those limitations landings will be cleared to the lowest possible altitude consistent with their position and distance from the runway.

Obstructions on the ground can be taken into consideration by imposing lowest safe altitudes along each leg in the approach structure. Maximum altitudes along the ground track will also be imposed. These have to be consistent with typical descent rates for each aircraft type. The upper altitude limits will, therefore, depend on the aircraft type. Unlike lower altitude limits, upper limits are not associated with a

specific location within the terminal area. Instead, they are related to the time left until touchdown. This means that the upper altitude limits applicable to any aircraft will have to be redetermined after a substantial change in its STAR. The reverse may also occur: on rare occasions, the current altitude of an aircraft may affect the aircraft's EFTAR.

5.5 Conflict Identification and Resolution

As stated in section 5.1, in order for the automated ATM/C system to operate properly, flight plans must be conflict-free. By this we mean that if all aircraft adhere to their flight plans within the specified conformance limits, the required airborne separations will be maintained at all times. An aircraft may be in conflict not only with other traffic, but also with ground obstructions and with restricted airspace. In terminal areas, airspace restrictions will usually result from weather conditions, (e.g. thunderstorms concentrated at particular regions of the terminal area).

Given the separation requirements and a set of flight plans, identification of conflicts is a straight forward task. Resolution of conflicts with ground obstructions and restricted airspace is also straight forward. In fact, as we pointed out in the previous section, altitude restrictions are inherent to the flight plan generation process. The same can be done with airspace restrictions.

Conflicts between two aircraft can be resolved by changing the flight plans of either (or both) aircraft involved. In order to determine which is the best choice, alternative flight plans for both aircraft have to be generated and evaluated. These may be in conflict with the flight plan of a third aircraft, and thus a new decision of the same type has to be made. This approach will require excessive computational effort and has to be rejected on that basis.

A much simpler conflict resolution logic is adopted. Flight plans are classified into three categories: invalid, temporary, and final. Invalid flight plans are those which do not conform to the schedule. This category includes aircraft that have been rescheduled, aircraft out of conformance, as well as aircraft that which have not yet been assigned flight plans. Temporary flight plans are in conformance with the runway schedule but have not yet been checked for conflicts. Final flight plans are conflict-free and will not change during current invocation of the flight plan generation function.

All three categories of flight plans have fixed portions. Usually, those will be parts of the flight plan which have been executed, or will be executed within a short time interval (say, 30-60 seconds) from the time the flight plan generation function was invoked. Additional parts of flight plans may be fixed due to constraints imposed by the ATM/C controller.

When the flight plan generation function is invoked, invalid flight plans are identified. At least one flight plan should be invalid for the function to be invoked. All valid flight plans are then made temporary. Finally fixed portions of all flight plans are identified.

Conflict checking is performed as follows:

1. Select an aircraft whose flight plan is temporary or invalid. If the flight plan is invalid, generate a valid one and designate it as temporary.
2. Identify conflicts with final flight plans as well as with fixed portions of invalid or temporary flight plans.
3. If no conflicts exist, make the flight plan final. Otherwise, generate a revised temporary flight plan and go back to step 2.
4. If no more aircraft to consider stop. Otherwise go to step 1.

In practice, the flight plan generation and the conflict resolution are performed simultaneously, i.e. as each the flight plan is generated for each phase of the approach, it is checked for conflicts and alternatives are generated if such conflicts are identified. If no conflicts exist, then flight planning continues with the next phase. When complete, the flight plan becomes final. This method guarantees that conflicts will be identified at the earliest possible time.

This approach to conflict resolution has obvious drawbacks compared with the more general approach summarized earlier in this section. It is possible, for example, to be unable to determine a set of conflict-free flight plans because of a bad choice in the sequence according to which aircraft are selected in step 1 above. It is, therefore, very important to determine a good criterion for making this selection.

We have chosen to select aircraft in an increasing order of their scheduled time of arrival at the runway. This implies that flight plans for aircraft which are close to landing become final first. As we have seen, flight planning flexibility decreases as the scheduled landing time approaches. Therefore, given two aircraft which are scheduled to land in direct succession, revising the flight plan of the second will, in most cases, be preferable to revising that of the first. This is exactly what the conflict resolution logic will do.

5.6 Separations during the Base and Final Approach Phases

When the scheduled time of arrival at the runway becomes fixed for some aircraft, it transitions from the adaptive to the precision stage of the approach. This will typically happen as the aircraft is flying along the assigned downwind leg.

Figure 5-4 shows an aircraft on the downwind leg (point P) and depicts a typical flight plan from there to the outer marker (OM). At

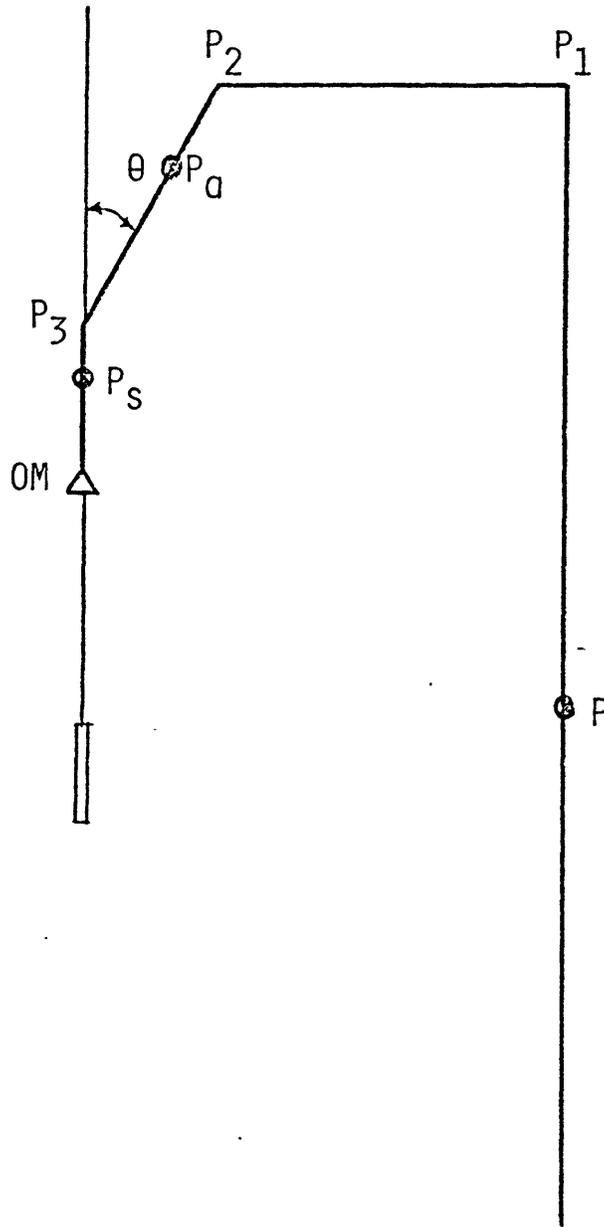


Figure 5-4. Typical Flight Plan for the Base and Final Approach Phases.

points P_1 and P_2 the aircraft turns to the base leg and to the runway intercept heading respectively. To allow easy interception of the runway centerline extension, angle θ is typically between 20 and 30 degrees. At the intercept point P_3 , the aircraft tracks the runway centerline extension towards the OM and subsequently towards the runway threshold.

Aircraft have to arrive at the OM not only at a specific time but also at a specific altitude and speed which will allow them to intercept the glideslope. We will call this altitude the OM crossing altitude. Typically, the OM is approximately 5 nautical miles from the runway threshold. This means that the OM crossing altitude is 1800 feet above ground level.

If the aircraft is not at the proper OM crossing altitude, an altitude command will be given (at some point P_a). Finally, the aircraft will be cleared to decelerate down to its final approach speed at some point P_s . P_a and/or P_s may occur either before or after point P_2 , and in any order.

This geometry is based on the work of Durocher [DUR 77] who showed that it can provide excellent accuracy in delivering aircraft to the OM. His tests were conducted by simulating approaches in a cockpit simulator. Radar tracking errors as well as wind were also simulated.

In this section we present a mathematical model that describes traffic patterns in the final approach region.¹ This model can be used to determine the separation between two aircraft at the time of closest approach, as a function of the flight planning parameters (e.g. the timing of the turn, the intercept angle θ , etc.), and the minimum required separations at the OM. Its purpose is twofold. First, it shows that safe separations can be maintained in the final approach region without any adverse effects on the precision of the delivery at the runway. Second, by studying the effects of the various flight planning parameters on the longitudinal separations, we develop guidelines which can be used to generate good traffic patterns in this area.

Good traffic patterns cannot be established once the aircraft is on the downwind or the base leg. At that point, all the flexibility still remaining in the timing of the turns and the final deceleration is needed to correct for navigation and surveillance errors. Planning for the desired traffic patterns in the final approach area has to start during the entry and the initial approach phase. The analysis in this section, therefore, is intended to provide guidelines for determining desired flight plans which are used for flight planning during the adaptive stage of the approach (see section 5.3).

(1) The model is similar to the one first used by Dunlay for estimating the expected number of conflicts at the intersection of airways.
[DUN 75]

Consider the situation of figure 5-5. Two aircraft are in the final approach phase. We will assume throughout that aircraft 1 is scheduled to land before aircraft 2, and that both aircraft have already decelerated to their respective final approach speeds, v_1 and v_2 .

Aircraft 2 is flying along the runway centerline extension. Aircraft 1 is going to intercept the runway centerline at an angle θ , and the intercept point P_i is at a distance d_i from the OM. We seek to determine the minimum separation, S_{\min} , between the two aircraft, as a function of:

1. v_1 and v_2 , their final approach speeds,
2. S_0 , the required minimum separation at the OM,
3. d_i , the intercept distance, and
4. θ the intercept angle.

If aircraft 2 reaches P_i before aircraft 1, the minimum horizontal separation will always be zero since, by assumption, aircraft 1 has to reach the OM first. We will, therefore, analyze the case where aircraft 1 reaches P_i first. To simplify the formulas, we will define P_i to be the origin of the coordinate system and we will assume that at time $t=0$ aircraft 1 is at P_i . Clearly, the direction of the flight is inconsequential. We will, therefore, assume that both aircraft are flying away from the OM.

The coordinates, (x_1, y_1) and (x_2, y_2) , of the two aircraft at any time t , are given by:

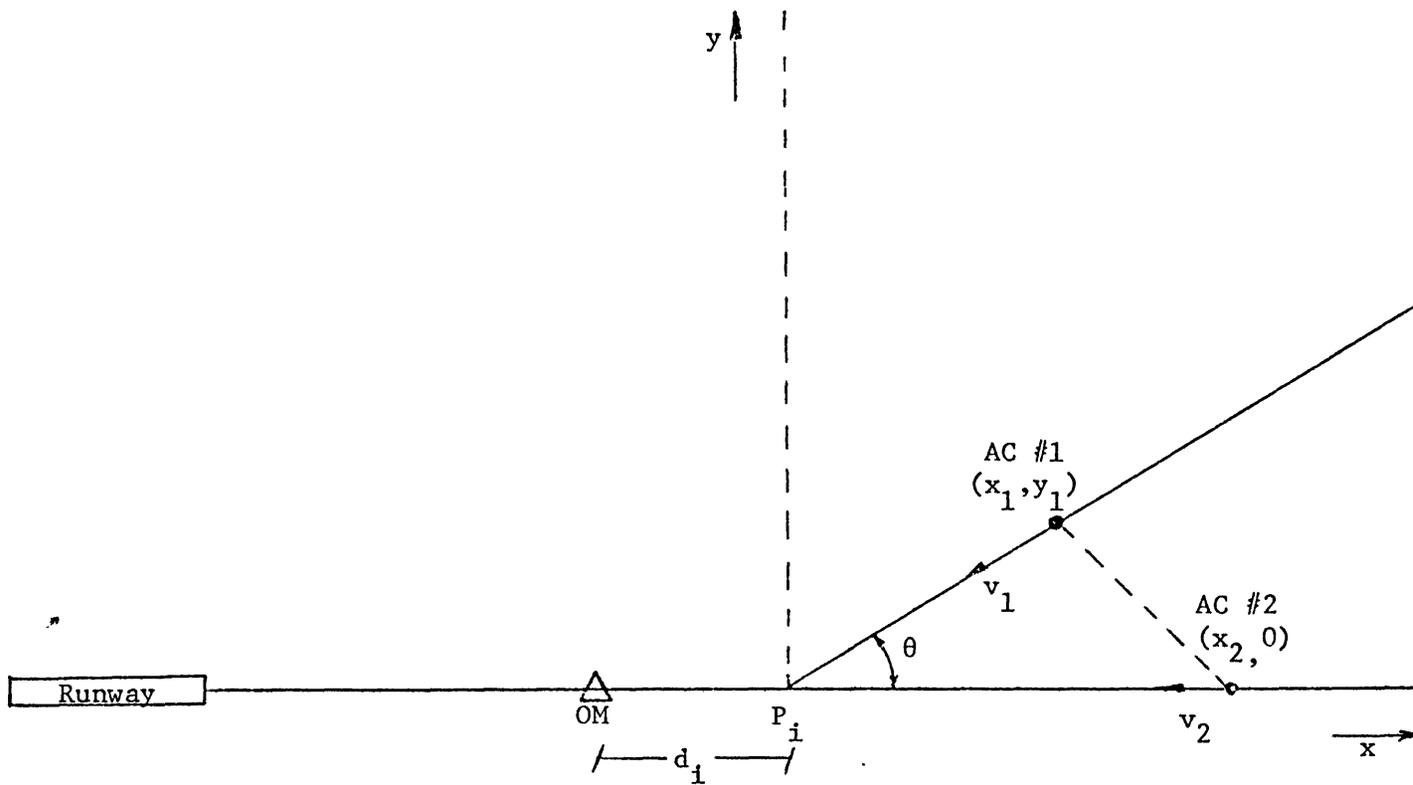


Figure 5-5. Horizontal Separations during the Final Approach Phase.

$$\begin{aligned} x_1 &= v_1 t \cos \theta & \text{and} & & x_2 &= S_i + v_2 t \\ y_1 &= v_1 t \sin \theta & & & y_2 &= 0 \end{aligned} \quad 5-1$$

where, S_i is the separation of the two aircraft at time $t=0$, i.e. when aircraft 1 is at P_i . When aircraft 1 is at the OM, the separation between the two aircraft is S_o . S_i is, therefore given by:

$$S_i = S_o - \frac{d_i}{v_1} (v_1 - v_2) \quad 5-2$$

By assumption, aircraft 1 reaches P_i first. This means that this discussion is valid for:

$$d_i < \frac{S_o v_1}{(v_1 - v_2)}$$

$S(t)$, the separation between the two aircraft as a function of time, satisfies:

$$S(t)^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

Substituting equations 5-1 for the aircraft coordinates we obtain after some manipulation:

$$\begin{aligned} S_2(t) &= t^2 [v_1^2 + v_2^2 - 2v_1 v_2 \cos \theta] \\ &+ 2t S_i [v_2 - v_1 \cos \theta] + (S_i)^2 \end{aligned} \quad 5-3$$

By differentiating with respect to time and setting the derivative to zero, we can determine the time t^* for which $S^2(t)$ (and therefore $S(t)$) is minimized. t^* is given by:

$$t^* = \frac{S_i(v_1 \cos \theta - v_2)}{[v_1^2 + v_2^2 - 2v_1 v_2 \cos \theta]} \quad 5-4$$

Equation 5-4 is valid only for positive values of t^* . Thus, we distinguish three cases.

First, if v_1 is less than v_2 , the minimum separation, S_{\min} , occurs at $t^*=0$ and is equal to S_i . Furthermore, according to 5-2, S_o is less than S_i . Thus, when v_1 is less than v_2 , the overall minimum separation occurs at the OM.

Second, if

$$v_1 \geq v_2 \geq v_1 \cos \theta$$

the minimum separation occurs again at $t^*=0$. This time, however, S_o is greater than S_i so the overall minimum separation occurs when aircraft 1 is at P_i and is given by 5-2.

Finally, if

$$v_1 \cos \theta > v_2$$

the minimum separation occurs at the time t^* given by 5-4, i.e. when both aircraft are beyond P_i . By substituting t^* from 5-4 into 5-3 and 5-1 we obtain the formulas for the minimum separation S_{\min} and for the

aircraft coordinates when at the time of closest approach. Specifically, the minimum separation is given by:

$$S_{\min} = S(t^*)$$
$$= \frac{S_i \sin \theta}{\sqrt{1 + r^2 - 2r \cos \theta}} \quad 5-5$$

where, r is defined as the ratio v_2/v_1 of the aircraft speeds.

Clearly, conflicts will arise during the final approach leg only when a slow aircraft (aircraft 2 in this case) follows a faster one. When this happens, altitude separation will have to be imposed in order to achieve tight separations at the OM. Fortunately, since fast aircraft followed by slow ones also contribute to inefficiencies in runway utilization, such pairs will occur less frequently when the runway schedule is optimized, than they would if aircraft were allowed to land in their random first-come-first-served sequence. In addition, takeoffs will normally be scheduled ahead of the following slow aircraft.

The fact that the optimal runway schedule will, on the average, have a smaller number of fast-slow aircraft pairs in direct succession, is not coincidental. The same principles are involved in both cases. The separations, however, that have to be considered by the runway schedule, occur between the OM and the runway threshold. Since altitude separation is not an option there, fast-slow sequences result in idle

runway time. The formulas for the separation requirements between the OM and the runway threshold are derived in Appendix A.

The indicated method for handling fast-slow sequences is to maintain altitude separation by clearing the fast aircraft down to the OM crossing altitude early and keeping the slow aircraft 1000 feet higher. The descent of the slow aircraft can be timed so that it reaches the OM crossing altitude when the fast aircraft is at the OM and the separation is S_o .

Table 5-1 illustrates the separations and the timing involved in this strategy. In cases 1 through 6, v_2 ranges from 140 to 90 knots while the other parameters are kept constant at the following values:

$$S_o = 3 \text{ nautical miles}$$

$$d_i = 2 \text{ nautical miles}$$

$$v_1 = 150 \text{ knots}$$

$$\theta = 20 \text{ degrees}$$

Case 7 is the same as case 6 except that S_o is changed from 3 to 6 nautical miles.

In addition to the value of the minimum separation, S_{min} , the following values appear in table 5-1:

1. v_2 : the speed of the slow aircraft,

TABLE 5-1

The effect of v_2 on Horizontal Separations
and on the Timing of the Descent

CASE #	v_2 (knots)	S_{\min} (NM)	t_1 (mins)	t_2 (mins)	x_1 (NM)	y_1 (NM)	x_2 (NM)	t_a (mins)	S_a (NM)
1	140	2.87	0.80	2.09	2.00	0.00	4.87	3.29	3.04
2	130	2.67	1.45	2.80	3.53	0.55	6.14	3.35	2.65
3	120	2.44	1.86	3.37	4.50	0.91	6.73	3.51	2.43
4	110	2.11	2.07	3.71	5.00	1.09	6.80	3.64	2.12
5	100	1.82	2.13	3.93	5.13	1.14	6.55	3.80	1.83
6	90	1.56	2.07	4.08	5.00	1.09	6.12	4.01	1.58
7*	90	3.69	3.84	7.84	7.50	2.06	11.76	6.00	4.31

* $S_0 = 6$ Nautical miles for case 7.

2. x_1 , y_1 , and x_2 : the coordinates of the two aircraft at the time of closest approach,
3. t_1 : the interval between the time of closest approach and the time aircraft 1 reaches the OM.
4. t_2 : the interval between the time of closest approach and the time aircraft 2 reaches the OM.
5. t_a : the interval between the time aircraft 2 starts its descent and the time it (i.e. aircraft 2) reaches the OM.
6. S_a : the separation between the two aircraft at the time aircraft 2 starts its descent.

Note that x_1 and x_2 are now relative to the OM and not to P_i . Also note that t_1 is equal to t^* since the analysis assumed that aircraft 1 was at P_i at time $t=0$.

The timing of the initiation of descent by the second aircraft was based on an assumed 500-600 feet per minute descent rate and an altitude difference of 1000-1200 feet. Accordingly, the descent is initiated 2 minutes before the first aircraft reaches the OM.

By comparing the values of t_2 and t_a , we note that, in cases 3 through 6, the descent of the second aircraft starts approximately at the time of closest approach. In case 7, the descent starts almost 2 minutes after the closest approach, and the corresponding separation at time t_a is 4.3 nautical miles. In cases 1 and 2 the descent has to start while the aircraft are still approaching each other. In both

cases, however, the minimum approach is almost three nautical miles. This means that, for all practical purposes, the aircraft have adequate longitudinal separation throughout.

From table 5-1 we see that, as expected, S_{\min} decreases as the difference between v_1 and v_2 increases. This result can be used to formulate a strategy for the timing of the final deceleration which is going to take place on this leg. Namely, we can maximize the separation at the time of closest approach by allowing the slow aircraft to decelerate near the OM while the fast aircraft decelerates immediately after the turn to the final approach leg. An extreme case of this would be if the slow aircraft decelerates from a speed v_2' to its final speed v_2 when the fast aircraft is at the OM. Then the closest approach would be the one applicable to the speed pair v_1, v_2' rather than the pair v_1, v_2 .

As we have said earlier, the timing of the final deceleration is used to provide the required accuracy in the delivery at the runway. Consequently it will not always be possible to implement the above strategy. It is possible, however, to plan so that on the average fast aircraft decelerate early and slow aircraft decelerate late. In particular, this can be achieved by appropriate selection of the desired flight plans (see section 5.3). The selection of the turn to the base and final approach legs can also be used to achieve this goal.

The general guidelines that can be deduced by the above analysis are:

First, the tight separation, S_0 , at the OM will require smaller longitudinal separations to occur in the final approach region whenever a slow aircraft is scheduled to land directly behind a faster one. Consequently, altitude separation will be required in order to assure safe separations for these fast-slow sequences. When such sequences exist in the runway schedule, it is imperative to maintain the slow aircraft at least 1000 feet above the OM crossing altitude until the last few minutes of the approach. Note the "laddering" of aircraft that will occur when n aircraft with speeds

$$v_1 > v_2 > \dots > v_n$$

are scheduled in direct succession. The third aircraft will have to remain 1000 feet higher than the second, the fourth 1000 feet higher than the third, etc. The operation on the base and final approach legs will then be very similar to the operation of a holding stack. As the leader aircraft reaches the OM, the second aircraft is cleared to the OM crossing altitude and all other aircraft in the sequence are cleared to 1000 feet below their present altitude.

Second, the runway centerline intercept point (P_i in figure 5-5) should be close to the OM for at least one of the aircraft. Furthermore, if the two aircraft are approaching from a different side of the runway, both should intercept as close to the OM as possible. It

can be easily shown from equation 5-5, that this will effectively increase the separation, S_{\min} , at the time of closest approach by increasing the angle θ between the two aircraft paths.

Third, the fast aircraft should decelerate down to its final approach speed as early as possible, while the slow aircraft should decelerate late. Again this will increase S_{\min} by making the velocity difference between the two aircraft smaller.

We conclude this section by noting that the calculations are conservative since the second aircraft will be at the OM crossing altitude 3 miles before it reaches the OM (6 miles for case 7). Even though landings typically intercept the glideslope at the OM, this need not always be the case. Instead, the second aircraft may intercept at its original altitude and be on the glideslope when it reaches the OM. Under this strategy, the separations between the two aircraft, at the time the second starts its descent along the glideslope, will be much larger. In fact, if S_0 , the required separation at the OM, is 3 nautical miles or larger, the two aircraft will maintain altitude separation at all times, independent of their difference in final approach speeds. The reason for this is, of course, that since the glideslope angle is approximately 3 degrees, 3 nautical miles of horizontal separation translate to an altitude difference of approximately 1100 feet.

5.7 Aircraft Guidance

Having generated flight plans for all the aircraft within the TCA, there still remains the problem of guidance. In the future, it is expected that the great majority of the commercial airline fleet will have advanced navigation and flight control systems which will allow them to conform to 4-dimensional flight plans with little or no help from the automated ATM/C system. Consequently, we will concentrate here on guidance issues that arise with respect to aircraft that do not have this capability.

There are two basic problems that we need to investigate. The first is the execution of turns from one leg to the next. The second has to do with speed control. Namely, do we have to exercise strict speed control throughout the approach, or can we allow the aircraft crew some flexibility in controlling the speed?

5.7.1 Recovery from Errors in the Execution of Turns

Let us consider the execution of a turn. Figure 5-6 shows an aircraft currently located at point P. The aircraft's flight plan calls for a turn to a different leg at point P_1 . If we assume, for the moment, that we have perfect information on the wind vector, W , on the the airspeed, v_a , of the aircraft, we can calculate the heading change required to track the new leg. Since the aircraft has a finite rate of turn, the time to initiate the turn is also of interest. Assuming a

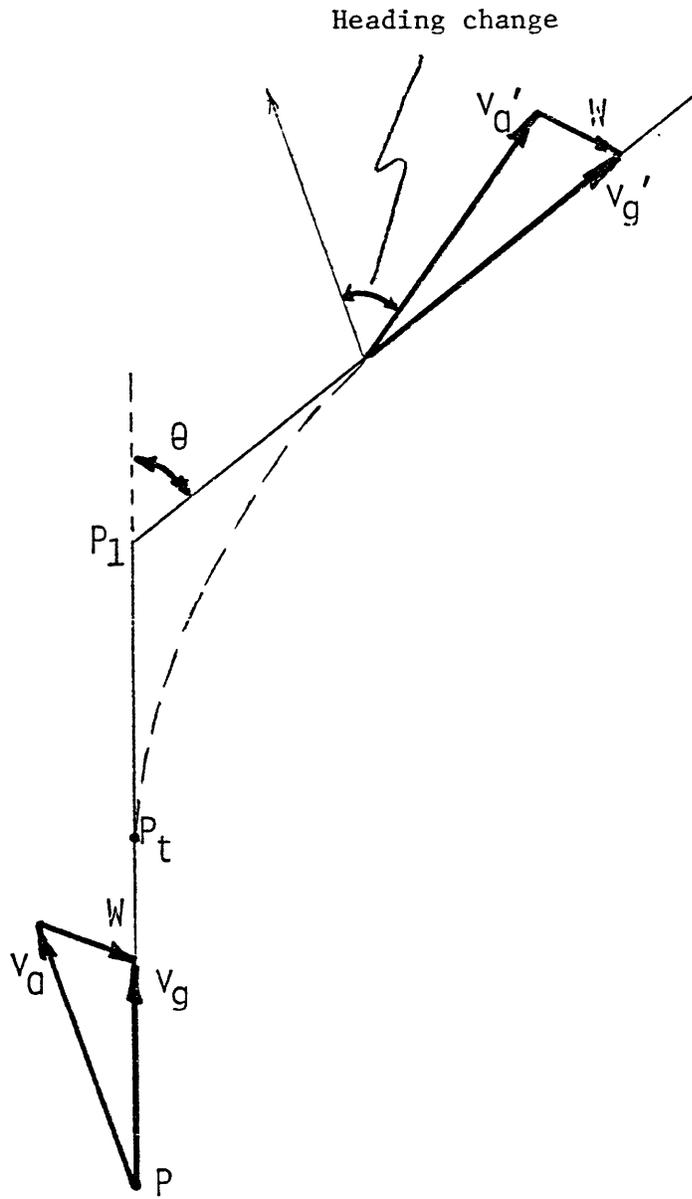


Figure 5-6. The Kinematics of Aircraft Turns

standard rate of turn, the actual aircraft track will be the arc of a circle (shown dashed in figure 5-6). We can, therefore, determine the point P_t (and therefore the timing) where the turn has to be initiated. The heading change and time to initiate the turn are the values required by the Command Processor for guiding conventionally equipped aircraft along their assigned track.

When surveillance errors as well as errors in the estimates of W , and v_a are introduced, inaccuracies will result in two ways:

First, the required heading change and the time to initiate the turn can no longer be accurately determined. As a result, the aircraft will generally be out of conformance with its flight plan after the turn. Lateral deviations from the intended track will be caused by early or late turn initiation. These may increase with time since the new ground track vector will not be parallel with the intended path. Longitudinal deviations will also be present and will generally increase with time.

Second, during the turn, the quality of the tracking data produced by the surveillance and tracking algorithms are significantly inferior to the position estimates that can be obtained while the aircraft is flying along a straight line. Consequently, the accuracy in executing the turn cannot be ascertained until good quality tracking data is again available. Typically, 30-40 seconds (i.e. 8-10 radar "hits") will be required after the aircraft comes out of the turn for the tracking

errors to be reduced to the level expected for linear flight. This time interval translates to approximately 3-5 nautical miles at the typical terminal area airspeeds for jet aircraft.

The degradation of tracking quality during a turn, requires all the legs of the flight plan to be at least 3-5 nautical miles, in order to be able to recover from the errors introduced by one turn, before the next turn is initiated. This restriction is of particular importance for the base and final approach legs, which are typically of the order of 5 miles in length. Fortunately, the proximity of the aircraft to the radar antenna (assumed to be located at the airport) provides better tracking accuracy at that stage of the approach.

It is clear from the above discussion that, after a turn either the flight path or flight plan of the aircraft will require correction. Large lateral errors can be corrected by the Conformance Resolution Path Generator. Typically the correction will take the form of a small heading change which will bring the aircraft closer to its assigned path. Even after the heading correction has been given, however, the path will not be the one originally intended by the flight plan. Normally this deviation will not create any conflicts. If it does, new flight plans will have to be generated for the conflicting pair of aircraft.

Small residual deviations from the flight plan can be corrected by the Command Processor prior to a subsequent turn. Longitudinal

deviations can be corrected by adjusting the time for the speed change, if one is specified in the flight plan for this leg. Lateral adjustments do not need correction, as long as they do not create a conflict. They have to be taken into consideration, however, in calculating the heading change and the timing of a subsequent turn.

Finally, if the deviations cannot be corrected by the Command Processor, new flight plans will have to be generated.

5.7.2 Speed Control

By necessity, 4-dimensional flight plans have to specify speed changes at various time-points. Nevertheless, this does not imply that speed needs to be actively controlled at all times. The proposed methodology uses ground track selection as the primary means of controlling timing at the runway. Active speed control is only used when it is absolutely necessary. From an operational standpoint, this simply means that, even though the flight plan may specify a speed reduction at some time-point, the Command Activator does not transmit that command to the aircraft.

An example will help illustrate how this can be accomplished. Assume that an aircraft is at point P of the downwind leg (figure 5-7) and that its current flight plan requires the aircraft to fly the ground track specified by the sequence of points P, P₁, P₂, P₃, OM with speed reductions at points S₁ and S₂. Furthermore, suppose that the aircraft

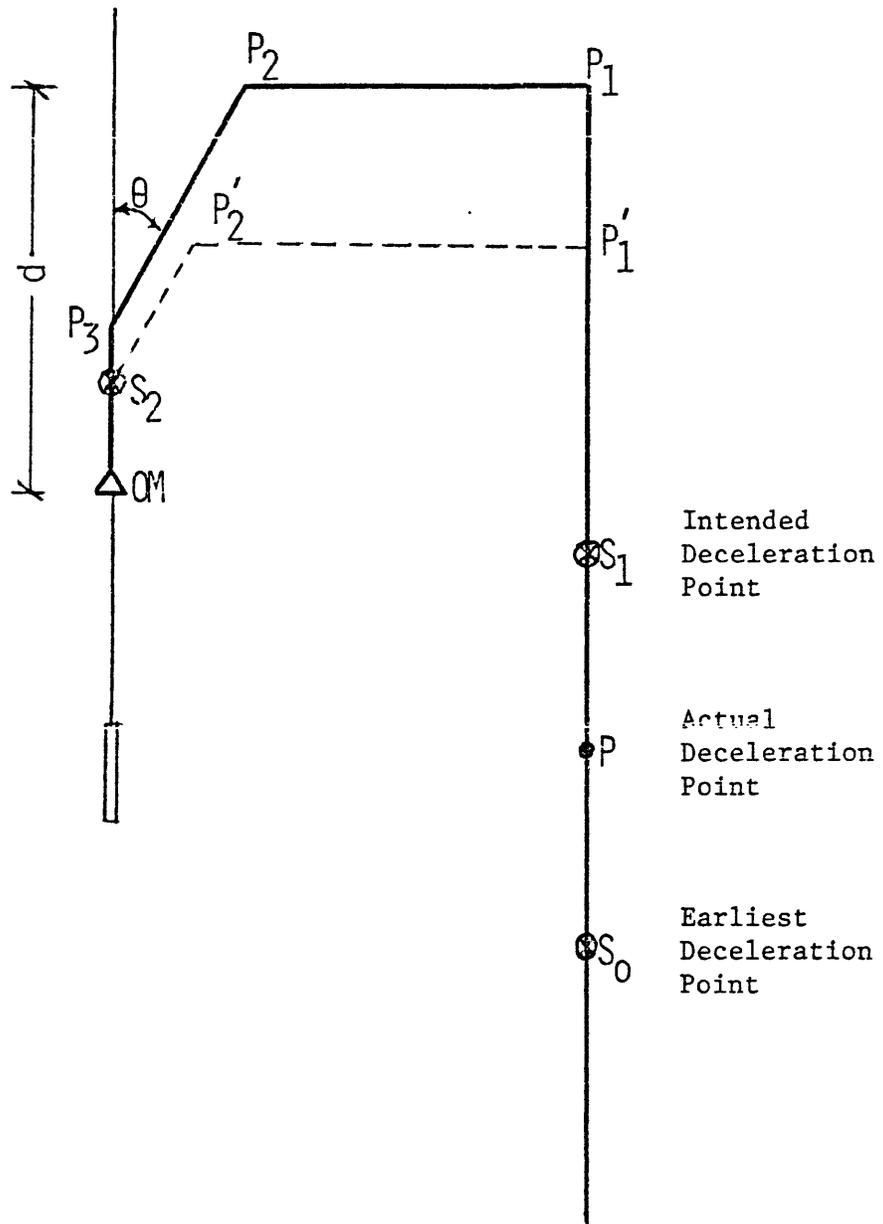


Figure 5-7. Recovery From Early Speed Reduction

decelerates unexpectedly at point P. The deceleration will cause the aircraft to fall out of conformance and a conformance alert will be generated. The only way to bring the aircraft back into conformance with the current flight plan will be to command a speed increase. Clearly this is not desirable. Instead, the flight plan is corrected by assigning a new base leg, thus shortening the length the aircraft has to travel to reach the runway. The new flight plan, therefore, be the one depicted in the dashed line.

There are limits to how early the pilot can reduce the airspeed. If the speed reduction occurs too early the aircraft will not be able to reach the runway on time, even if the shortest available 2-dimensional path is assigned. Such situations can be avoided by determining in advance the point prior to which the flight plan generator cannot compensate for a speed reduction. As an example this may be point S_0 in figure 5-7. S_0 will depend on the distance, d , of the currently assigned base leg from the OM, as well as the range of allowable values for that distance.

The general method for avoiding active speed control can be summarized as follows: At the time of a turn to a new leg of the approach path, the point S_0 , beyond which the flight plan generator can compensate for speed reductions, is determined. The pilot is then advised to maintain his speed until that point. Once the aircraft reaches S_0 , the pilot is free to reduce his speed at his discretion and

the flight plan generator will compensate for it by adjusting the point P_1 at which the aircraft will turn to base.

Note that the same method can also be used if the aircraft maintains its current speed longer than specified in the flight plan. Finally, as we have mentioned earlier, this method cannot be used on the final approach leg. The final speed reduction has to be precisely controlled to achieve accurate delivery at the OM.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

The purpose of this research was to study decision making process in the terminal area Air Traffic Management and Control (ATM/C) system, and develop the methodology for the design and implementation of a fully automated terminal area ATM/C system. Our work adopts the "centralized decision making" methodology which is practiced at the present time. Namely, the decision-making authority rests with the ATM/C controller. Accordingly, the main focus of our work is the Ground Control Subsystem, (AGCS), of the terminal area ATM/C system. Other subsystems (e.g. the aircraft control subsystem, and the various data link systems) are considered to the extent that they affect the decision-making process in the AGCS.

In summary, the accomplishments of this research were in 4 areas:

First, we studied the Automated Ground Control Subsystem, (AGCS), at a top level of aggregation, identified its elements, described their functional requirements, and studied their interrelationships. From this functional description of the AGCS, emerged the concept as well as the top level software design for the automated terminal area ATM/C system, which provided the framework for the remainder of our research.

Second, we developed a very general formulation of the runway scheduling problem and examined the factors that contribute to its complexity. Furthermore, we examined in detail the interactions of runway scheduling with the ATM/C controller as well as with the remaining automation functions. Finally, we designed and implemented a heuristic algorithm for generating efficient runway schedules. The algorithm is based on the work of Dear, [DEA 76], and uses the Constrained Position Shifting methodology that he proposed. The algorithm is capable of scheduling aircraft on multiple runways and was specifically designed to be implemented in a real-time environment.

Third, we developed a methodology for generating conflict-free, 4-dimensional flight plans. Our methodology is designed to generate flight plans that can be easily modified to adapt to a dynamically changing runway schedule. In connection with this methodology, we discussed the problem of aircraft guidance and showed how our flight planning methodology can be used to reconcile precise 4-dimensional flight planning on the part of the automated ATM/C system, with conventional navigation capability onboard the aircraft.

Fourth, we designed and implemented a real-time terminal area simulation facility to be used as a testbed for further research, development and testing of automation software. The design of the software is based on our concept for the automated terminal area ATM/C system. ATM/C automation is a vast area of research and requires a "critical mass" of software before experimentation and testing can

begin. We feel that this work has significantly contributed towards the achievement of this critical mass. Furthermore, the software was specifically designed to facilitate further development of ATM/C automation functions. In particular, the simulation facility has the following characteristics:

1. Modular software design which allows easy modification of specific simulation functions with little or no effect on others. In particular, the logic of the Simulation Monitor process can control any number of subordinate processes. This allows multiple controller positions to be simulated. In addition, pseudo-pilot stations may be added, allowing direct control of the simulated aircraft by human operators. Pseudo-pilot stations will add an extra degree of realism in the simulation and may prove necessary particularly for "human factors" research. Finally, the concept of timer loops and the fact that any number of them may be implemented, allows great flexibility in modifying the flow of events in the simulation. For example, tracking and surveillance data are now updated once every four seconds for all aircraft. In research which is specifically interested in tracking algorithms, this may not be satisfactory since the timing of the surveillance data, which depends on the aircraft bearing may be of importance. The simulation logic allows

such modifications to be made with relatively little effort.

2. The simulated aircraft motion is based on detailed models of the aircraft control system. Errors in the navigation and cockpit instrument readings are also modelled in detail. This results in a very realistic representation of aircraft flight quality as seen by the ATM/C controller. Proper modelling of aircraft flight is particularly important in the study of advanced terminal area Air Traffic Control since the precision with which aircraft can be guided along 4-dimensional flight plans has tremendous effect on the flight planning logic.
3. The interface between the controller and the automation functions was specifically designed to allow experimentation and testing of alternative display concepts. This was done by structuring the software in two "layers". The lower layer manages the proper update of the screen and is independent of the actual data that is displayed. Thus the displayed data as well as the display format can be changed by modifying only the top layer of the I/O interface.

6.2 Recommendations for Future Research

There are two main areas where research and development is still needed before a successful implementation of an automated terminal area ATM/C system is accomplished:

1. Algorithms for Traffic Flight Plan Generation

The Metering and Spacing program has shown that automatic flight plan generation can be achieved. The current test version, however, has significant drawbacks which we have pointed out (chapter 1). Most importantly, the current software may be totally incompatible with the runway scheduling function which is expected to provide most of the improvement over the current ATM/C system capability. The methodology for automated flight plan generation proposed in this work should provide a solid basis for further work in this area.

From our experience with the subject we believe that flight plan generation will be a particularly fertile ground for applications of Artificial Intelligence methods. Specifically, the conflict resolution logic can be a "learning program" which accumulates experience regarding the selection of alternative flight plans when conflicts are identified.

2. Human Factors research and experimentation

The introduction of automated runway scheduling and flight plan generation will substantially increase the already large amount of information available to the ATM/C controller. This creates first the need for experimenting with alternative methods of presentation. The interaction of the ATM/C controller with the automation software is of course the other significant area in human factors research. Finally, we discussed in chapter 4 the need to re-evaluate the allocation of responsibilities among various controller positions in the terminal area.

Independent of our views on the role of the controller in an automated ATM/C environment, we believe that in all three of the above areas there is a need for quick testing and evaluation of alternative concepts. TASIM can provide an effective tool for human factors research at the concept level due to the relative ease with which the display format and the I/O interface can be modified.

The existence of a precise runway schedule and of flight plans for all aircraft in the terminal area give rise to the possibility of improving the capabilities of a number of "established" functions in the ATM/C system. In chapter 2 we described possible improvements in hazard detection and tracking through algorithms which use flight plan as well as surveillance data. Additional improvements may be possible in both

these functions if we provide them with readings from the onboard equipment (e.g. heading, speed and vertical speed indicators, etc.). This can be achieved through the use of the air-to-ground digital data link for mode-S equipped aircraft.

Finally in chapter 4 we discussed the connection between runway scheduling and congestion management or flow control both at the entry fixes and at the departure gates. Runway scheduling information can be used to allow much more efficient flow control in the enroute airspace. The runway schedule is therefore a valuable source of information for automated enroute flow control systems such as the one included in the concept description of AERA.

- blank page -

APPENDIX A

INTERARRIVAL DYNAMICS

This appendix presents the mathematical formulas for determining the minimum time separation between consecutive arrivals using the same runway. The calculations assume that the two basic ATC rules related to the runway operation are:

1. No two aircraft are permitted on the same runway at the same time.
2. Coaltitudinal aircraft under ground control must maintain a specified horizontal separation.

It is also assumed that all controlled aircraft arriving at the same runway fly a common final approach path at a constant velocity equal to the aircraft's preferred approach speed. The preferred approach speed depends upon such parameters as the type of aircraft, the landing weight, the weather conditions, and pilot preferences. This preferred approach speed is specified by the pilot when the aircraft arrives at the entry fix. Consequently, two identical aircraft may have different preferred approach speeds.

The minimum time separation at the runway between two successive landings is analytically determined as a function of the final approach

length, the approach speeds and the minimum horizontal separation distance. Specifically, let:

$v_{\text{land}}(i)$ = the approach speed of aircraft i

$t_{\text{occ}}(i)$ = the runway occupancy time of aircraft i

s_{ij} = the minimum horizontal separation for aircraft i followed by aircraft j

F = the length of the common final approach path.

Then, $t_{ij}(s_{ij}, F)$, the minimum time separation at the runway between the landing of aircraft i followed by aircraft j is given by:

$$\max[t_{\text{occ}}(i) ; s_{ij}/v_{\text{land}}(j)]$$

when $v_{\text{land}}(i)$ is less than $v_{\text{land}}(j)$, and by:

$$\max[t_{\text{occ}}(i) ; s_{ij}/v_{\text{land}}(j) + F(1/v_{\text{land}}(j) - 1/v_{\text{land}}(i))]$$

when $v_{\text{land}}(i)$ is greater than $v_{\text{land}}(j)$.

To simplify this expression, let ϵ_{ij} be defined as follows:

$$\epsilon_{ij} = \left\{ \begin{array}{ll} 0 & , \quad v_{\text{land}}(i) < v_{\text{land}}(j) \\ 1/v_{\text{land}}(j) - 1/v_{\text{land}}(i), & v_{\text{land}}(i) \geq v_{\text{land}}(j) \end{array} \right\}$$

then,

$$t_{ij}(s_{ij}, F) = \max [t_{occ}(i) ; s_{ij}/v_{land}(j) + F\epsilon_{ij}]$$

Figures A-1 and A-2 illustrate the two landing situations. The runway occupancy time of aircraft 1 is assumed to be less than $s_{ij}/v_{land}(2)$ in both cases.

In the overtaking situation of figure A-1, aircraft 2 is faster than aircraft 1 and consequently, the point of closest approach between the two aircraft occurs when aircraft the first one touches down on the runway. This closest approach equals $s_{12}/v_{land}(2)$.

In the opening case of figure A-2, aircraft 2 is slower than aircraft 1 and the point of closest approach occurs when aircraft 1 begins its final approach. Aircraft 1 lands $F/v_{land}(1)$ time units after beginning the final approach and aircraft 2 lands $(F + s_{12})/v_{land}(2)$ time units after aircraft 1 begins its final approach. Thus the time between the two landings is:

$$(F + s_{12})/v_{land}(2) = s_{12}/v_{land}(2) + F\epsilon_{12}$$

Suppose now that three aircraft, with approach speeds 120, 135, and 150 knots respectively, have identical preferred times of arrival at the runway, (arbitrarily set to 0). Table A-1 presents the minimum time separation for all possible combinations of successive arrival pairs. A final approach length of 5 nautical miles was assumed for the

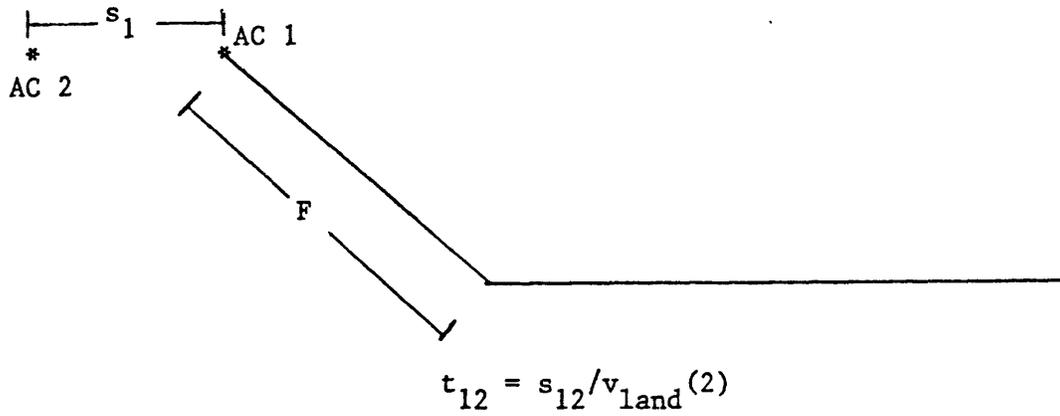


Figure A-1. Minimum Interarrival Separation (Overtaking Case)

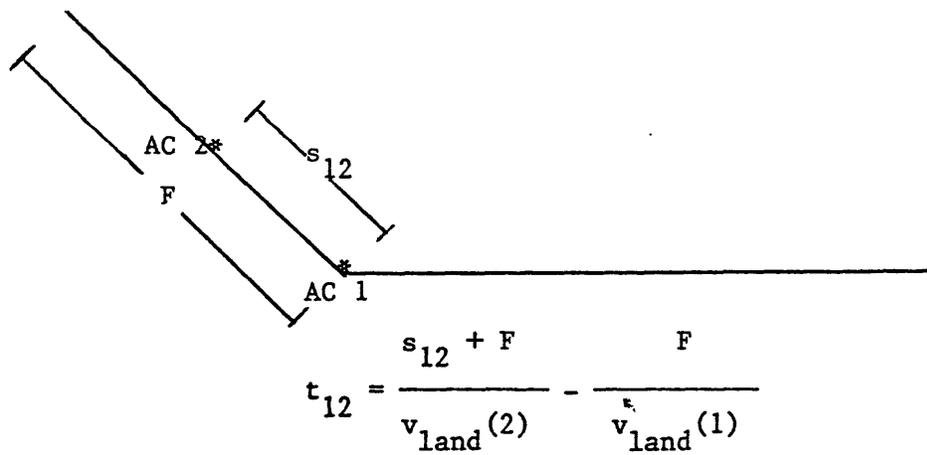


Figure A-2. Minimum Interarrival Separation (Opening Case)

TABLE A-1
Minimum Interarrival Time Separation (Seconds)
At the Runway

Leading
Aircraft
(Knots)

Following Aircraft (Knots)

	120	135	150
120		80.00	72.00
135	106.67		72.00
150	120.00	93.33	

$$s_{ij} = 3 \text{ NM}$$

$$F = 5 \text{ NM}$$

calculations. In this example, the runway occupancy times are irrelevant as long as they do not exceed 72 seconds.

There are six sequences in which the aircraft can land. Table A-2 lists the assigned landing times (rounded to the nearest second) for each landing sequence and the average delay experienced by the three aircraft. The delay for each aircraft is assumed to be the difference between the assigned landing time and the preferred landing time (which is zero in this case).

Several important points are brought out in this example. First, even in this simple situation, the sequence of operations greatly affects the runway utilization as demonstrated by the range in the time the last aircraft lands (152 vs. 200) and the variation in the average delay (74 vs 106.67). Second, the fact that some sequences have almost the same average delay (e.g. sequences 3 and 6) or the same last landing time (e.g. sequences 1 and 2) is not coincidental. Such situations occur very often and indicate that it is necessary to consider more than one efficiency measure in optimizing the runway schedule.

TABLE A-2
A Comparison of Landing Sequences

<u>Seq. 1</u>	<u>Landing Time</u>	<u>Seq. 2</u>	<u>Landing Time</u>	<u>Seq. 3</u>	<u>Landing Time</u>
1	0	1	0	2	0
2	80	3	72	1	107
3	152	2	165	3	179
Average Delay:	74	Average Delay:	75.67	Average Delay:	95.33
<hr/>					
<u>Seq. 4</u>	<u>Landing Time</u>	<u>Seq. 5</u>	<u>Landing Time</u>	<u>Seq. 6</u>	<u>Landing Time</u>
2	0	3	0	3	0
3	72	1	120	2	93
1	192	2	200	1	200
Average Delay:	88	Average Delay:	106.67	Average Delay:	97.67

- blank page -

APPENDIX B

THE KINEMATICS OF THE INTERCEPT VECTOR

The kinematics of determining a vector to cause the intercept of a moving target can be expressed as a function of the relative speeds and the relative position of the aircraft and the target. To simplify this discussion, turning radii, the time to turn and the time to decelerate will be ignored. These complicate unnecessarily the basic concept and can be accounted for by calling turns and speed changes earlier than the idealized model requires.

Consider figure B-1. At time $t=0$ the target box is at the origin and is moving along the x axis at a constant speed v_b . The aircraft is at a point (x_0, y_0) and has a ground speed v_0 . This speed may be equal to the aircraft's approach speed v_a . The intercept problems can be stated as:

1. Is it possible to intercept the box?
2. What heading θ is required to intercept?
3. What is the time t_v to call the speed change if v_0 is greater than v_a ?
4. What is the time to intercept, t_i ?

Since it is obvious that the required heading θ depends on the time the speed change is called, we can set the time to call that speed change at $t_v = t_i/2$. The average aircraft ground speed in the interval

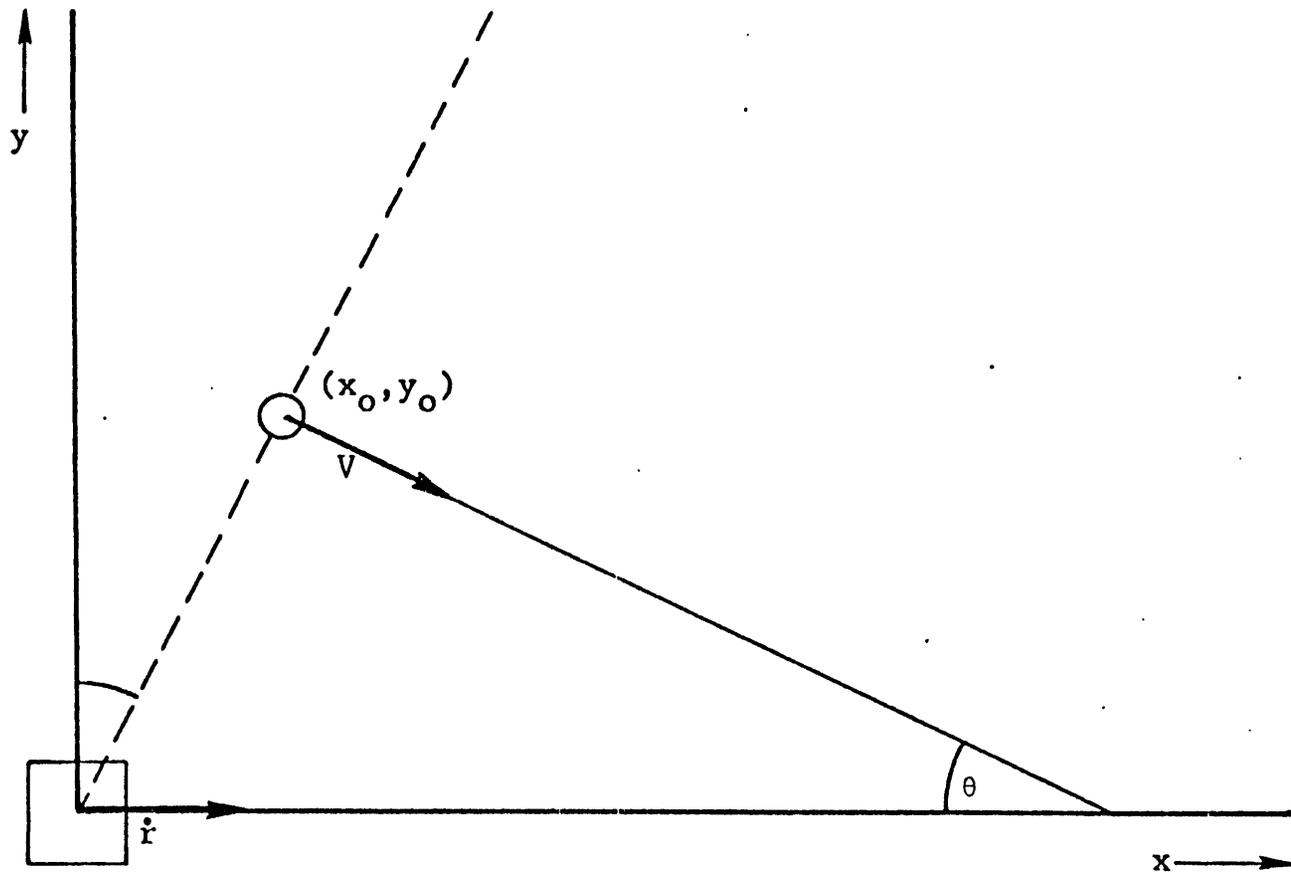


Figure B-1. The Kinematics of the Intercept Vector

(t_0, t_i) will be $v = (v_0 + v_a)/2$. This assumption allows some flexibility in correcting a turn that was not perfectly timed by adjusting the time for the speed change after the turn is completed and good tracking of the aircraft has resumed.

Accordingly, we can assume that the aircraft travels at a speed v from time t_0 until it intercepts the box at time t_i .

The equations of motion are:

$$r = \dot{r}t$$

$$\dot{x} = v \cos \theta \quad x = x_0 + vt \cos \theta$$

$$\dot{y} = v \sin \theta \quad y = y_0 - vt \sin \theta$$

At the time of intercept, $y = 0$ and $x = r_i = \dot{r}t_i$. Thus,

$$y_0 = vt_i \sin \theta \quad \text{B-1}$$

$$\dot{r}t_i = x_0 + vt_i \cos \theta \quad \text{B-2}$$

From B-1, the time of intercept is given by:

$$t_i = \frac{y_0}{v \sin \theta}$$

From B-2, substituting for t_i ,

$$\frac{\dot{r} - v \cos \theta}{v \sin \theta} = \frac{x_0}{y_0}$$

or

$$\frac{\dot{r}}{v} = \cos \theta + \frac{x_0}{y_0} \sin \theta \quad \text{B-3}$$

From B-3, if we know the ground speed ratio \dot{r}/v , and the intercept angle θ , we can solve for the ratio x_0/y_0 . Note that only the ratio is of interest and not the actual values of x_0 and y_0 . If we denote by

$$\alpha = \tan^{-1} \left(\frac{x_0}{y_0} \right)$$

the relative bearing of the aircraft from the box, for any given values of the ratio \dot{r}/v and of θ , the box can be intercepted as long as the aircraft is on the dotted line of figure B-1. The time to intercept the box t_i , and the point of intercept on the x axis, will depend on the position of the aircraft along the dotted line.

The relative bearing, α , determines the position of the wand on the final vectoring display discussed in section 4.11.

REFERENCES

- AHO 74 Aho, A. V., Hopcroft, J.E., and Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Welsey, Menlo Park, California, June 1974.
- ATH 71 Athans, M. and Porter, L. W., "An Approach to Semi-Automated Scheduling and Holding Strategies for Air Traffic Control," 1971 Joint Automatic Control Conference Preprints, August 11-13, 1971.
- BLU 60 Blumstein, A., An Analytic Investigation of Airport Capacity, Ph.D. Thesis, Cornell University, 1960.
- BRA 77 Bradley, S. P., Hax, A. C., and Magnanti, T. L., Applied Mathematical Programming, Addison-Wesley, Reading, Ma., 1977
- COU 78 Couluris, G. J., Tashker, M. G., and Penick, M. C., Policy Impacts of ATC Automation: Human Factors Considerations, Report No. FAA-AVP-78-1, DOT-FAA, Office of Aviation Policy, Washington, D.C., January 1978.
- CRE 73 Credeur, L. et al., Study of the Impact of Air Traffic Management Systems On Advanced Aircraft and Avionic Systems, NASA, Langley Research Center, CR-132278, Hampton, Virginia, 1973.

- DEA 76 Dear, R. G., The Dynamic Scheduling of Aircraft in the Near Terminal Area, FTL Report R76-9, Flight Transportation Laboratory, M.I.T., Cambridge, Ma, August 1976.
- DRA 67 Drake, A. W., Fundamentals of Applied Probability Theory, McGraw-Hill, New York, 1967.
- DUN 75 Dunlay, W. J. Jr., "Analytic Models of Perceived Air Traffic Control Conflicts," in Transportation Science, Vol. 9, No. 2, May 1975, pp. 149-164.,
- DUR 77 Durocher, C. L., Ground Controlled Precision Landing Delivery in the Presence of Radar Disturbances, FTL Report R77-2, Flight Transportation Laboratory, M.I.T., Cambridge, Ma, May 1977.
- GOL 81 Goldmuntz, L., et al., Automated En-Route Air Traffic Control, Report # FAA-EM-81-3, FAA, Office of Systems Engineering and Management, Washington D.C., March 1981.
- HEI 76 Heinz, V., Stochastic Simulation of Terminal Area Airspace, S.M. Thesis, M.I.T. Department of Aeronautics and Astronautics, Cambridge, Ma, September 1976.
- HOC 74 Hockaday, S., and Kanafani A., "Developments in Airport Capacity Analysis", in Transportation Research, Vol. 8, pp. 171-180, 1974.

- HOF 72 Hoffman, W. C., Hollister, W.M., and Simpson, R.W., Functional Error Analysis for ATC Concepts Evaluation, DOT-TSC-212-72-1, Aerospace Systems Inc., Burlington, MA., May 1972.
- HOL 74 Holland, F. C., and Garceau, T. V., "Geneology of Terminal ATC Automation," Revision 2, The MITRE Corporation, March 1974.
- HSI 76 Hsin, C. C., An Analytical Study of Advanced Terminal Area Air Traffic Management and Control, FTL Report R76-13 Flight Transportation Laboratory, Cambridge, Ma, September 1976.
- MAT 77 System Software Specification, Marine Air Traffic Control and Landing System (MATCALs), Systems Engineering, Vallejo, Ca, September 1977.
- ODO 69 Odoni, A. R., An Analytical Investigation of Air Traffic Control in the Vicinity of Terminal Areas, M.I.T. Operations Research Center, Technical Report No. 46, December 1969.
- PAP 82 Papadimitriou, C. H., and Steiglitz, K., Combinatorial Optimization: Algorithms and Complexity, Prentice-Hall, Englewood, N.J., 1982.
- PAR 79 Pararas, J. D., Sequencing Tasks on a Multi-Processor Facility, M.I.T. Flight Transportation Laboratory, FLT-Memo 78-2, September 1978.

- PAR 81 Pararas, J. D., Simpson, R. W., and Hoffman, W. C., Development of a Terminal Area Simulation for Marine Air Traffic Control and Landing System, Flight Transportation Associates, Interim Technical Report FTA-TR-81-201, Cambridge, Ma, January 1981.
- PAR 81a Pararas, J. D., An Airport Capacity Model, Flight Transportation Associates, Technical Report FTA-TR-117, Cambridge, Massachusetts, 1981.
- PAR 82 Pararas, J. D., et al., Computer Program Development Specification for the ATC Subsystem of the MVSRE, M.I.T., Flight Transportation Laboratory, FTL Report R82-1, Cambridge, Massachusetts, January 1892.
- POR 79 Poritzky, S.B., "ATC Automation: the Dilemma of Automated Decisionmaking", Presented at ATCA, Atlantic City, N.J., October 1979.
- PSA 78 Psaraftis, H. N., Dynamic Programming Algorithms for Specially Structured Sequencing and Routing Problems in Transportation, Ph.D. Dissertation, Department of Ocean Engineering, M.I.T., Cambridge, Ma, September 1978.
- RAT 64 Ratcliffe, S., "Congestion in Terminal Areas," Institute of Navigation Journal, Vol. 17, April 1964, pp. 182-186.

- ROW 78 Rowland, G., "Terminal Conflict Alert," In SRDS Progress Report, DOT-FAA, Washington, D.C. August 1978.
- SAR 71 Sarris, A. H., and Athans, M., An Algorithm for Terminal Area Air Traffic Control, M.I.T., Electronic Systems Laboratory, Report ESL-P-466, December 1971.
- SCH 73 Schmidt, D. K., and Swain, R. L., "An Optimal Control Approach to Terminal Area Air Traffic Control," Journal of Aircraft, Vol. 10, No. 3, March 1973, pp. 181-188.
- SIM 64 Simpson, R. W., An Analytical Investigation of Air Traffic Operations in the Terminal Area, Ph.D. Dissertation, Department of Aeronautics and Astronautics, M.I.T., Cambridge, Ma, May 1964.
- SIM 66 Simonnard, H., Linear Programming, Prentice Hall, Englewood Cliffs, N.J., 1966.
- TAL 78 Talley, J. R., "Basic Metering and Spacing for ARTS III," In SRDS Progress Report, DOT-FAA, Washington, D.C., August 1978.
- TAL 80 Talley, J. R., "Basic Metering and Spacing for ARTS III," In SRDS Progress Report, DOT-FAA, Washington, D.C., 1980.
- WES 81 Wesson, R., et al, Scenarios for Evolution of Air Traffic Control, Prepared for the FAA by the Rand Corporation, Report No R-2698-FAA, Santa Monica, California, November 1981.