THE UNIVERSITY OF CALGARY

A FAST-TIME SIMULATION OF AIR-TRAFFIC CONTROL

by

James A. Inkster

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTER OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE

CALGARY, ALBERTA

MARCH, 1987

ISBN 0-315-46609-X

THE UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a thesis entitled "A Fast-Time Simulation of Air-Traffic Control" submitted by James A. Inkster in partial fulfillment of the requirements for the degree of Master of Science.

Brian Unger

Supervisor
Brian W. Unger, Ph. D.
Department of Computer Science

Graham Birtwistle

Dr. G. Birtwistle
Department of Computer Science

David Irvine-Halliday

Dr. D. Irvine-Halliday
Department of Electrical Engineering

(DATE) May 26, 1987

ABSTRACT

This thesis describes the design and implementation of a fast-time simulation model of air-traffic movement in the controlled airspace surrounding an airport. The model simulates the flow of arriving and departing air traffic, and may be used by airspace planners to estimate traffic capacity and flight delays under hypothesized traffic conditions, air-traffic control policies, and airspace configurations.

The model is written in the SIMULA language and uses discrete and continuous simulation techniques. Continuous components of the model approximate the vertical and horizontal movements of aircraft in accordance with aircraft performance data. Discrete components include an ATC process that maintains separation among aircraft to create a realistic traffic flow pattern. The results of a simulation are presented graphically in plots of delay and throughput statistics, and in an animated display of aircraft positions superimposed on an airspace map.

The thesis presents results for a typical fast-time simulation experiment, and discusses directions for future development of the model.

ACKNOWLEDGEMENTS

I acknowledge most gratefully the assistance given to me by the management and staff of the Research and Experimentation Centre of Transport Canada's Air Traffic Services. In particular, K.M. Duncan and Richard Young provided continuous technical guidance in the formulation of the model, and deserve much of the credit for the success of the project. Needless to say, deficiencies in the model, including misinterpretations of the principles of air-traffic control and air navigation, are my responsibility.

I am pleased to acknowledge the invaluable contribution of David Williamson of DMR Group Inc., who implemented most of the TFS software. As usual, his work was fast, accurate, and professional.

Finally, I am grateful to Dr. Brian Unger for his continued support of this thesis project throughout a long gestation, and to Dr. Graham Birtwistle, both for specific technical assistance and for his help in initiating the project at the Research and Experimentation Centre.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

The general problem addressed in this thesis is the computer simulation of air-traffic movement in controlled airspace for the purpose of estimating traffic throughput and delay under hypothesized traffic patterns, air-traffic control policies, and airspace configurations. In the specific work described here, a fast-time air-traffic simulation model was designed by the author and implemented under his direction, at the Research and Experimentation Centre operated by the Department of Transport's Air Traffic Services in Hull, Quebec. The current version of the model simulates the flow of arriving and departing air traffic in the controlled airspace surrounding an airport, and may be used by airspace planners to estimate traffic capacity and flight delays under a variety of hypothesized conditions.

As the following chapters demonstrate, the simulation of air-traffic movement is a demanding problem, requiring the application of continuous simulation methods to represent complex aircraft manoeuvres, as well as the discrete-event modelling of the air-traffic controller's problem-solving activities, and his communication with the aircraft under his control. Nevertheless, there is an immediate need for accurate and comprehensive air-traffic simulation models to serve the unique requirements of airspace planning. The following sections discuss the importance of simulation and modelling methods in airspace planning, and explain the particular role of fast-time simulation.

1.1 Simulation and Airspace Planning

Under the long-term system development and integration strategy defined in Canadian Airspace Systems Plan (CASP) by the Department of Transport (DOT), safe and efficient utilization of Canada's domestic airspace will be maintained not only by the on-going implementation of new technology but also by continued refinement of the airspace organization and of air-traffic control (ATC) procedures for controlling and sequencing aircraft in the airspace. The effective use of both existing and future ATC and air-navigation equipment will require the planning of new procedures that will improve the use of existing systems and optimize the use of the replacement systems.

Because of the obvious risks of experimentation in the actual airspace, air-traffic systems planning relies heavily on modelling and simulation techniques for the evaluation of proposed changes to airspace organization or procedures. At the DOT Research and Experimentation (R&E) Centre, real-time simulation has been used extensively for the development of new ATC procedures and airspace structures. An important component of CASP is the modernization of the Centre's simulation facilities, including the addition of a fast-time simulation capability to complement the real-time simulator.

1.2 The Role of Fast-Time Simulation

In real-time simulation, a computer-based radar simulator, controller display consoles, voice communication circuits, and simulated traffic flow are combined to create a realistic working environment for air-traffic controllers. Real-time simulation has proved to be extremely successful as a method of evaluating new ATC procedures and training controllers in the use of new procedures, but it is labour-intensive and expensive. Each simulation project requires the participation of a team of simulation specialists and air-traffic controllers to collect data, to prepare maps and procedures for the exercise, and to conduct the exercise itself. The cost of a real-time simulation exercise may include several months of work to prepare the exercise, as well as travel and living costs for controllers from regional control units who are participating in the exercise.

However, there are circumstances in which a less expensive, fast-time simulation would be appropriate, particularly when the focus of a study is on traffic-flow patterns and throughput, rather than on controller procedures and workloads. In fast-time simulation, components of the air-traffic system are modelled by computer programs that imitate selected aspects of the system's behaviour. Fast-time simulation does not require the participation of controllers, pilots, or technicians, and travel is limited to that required for field observation and data gathering.

In addition to lower costs, fast-time simulation has the advantage of speed. Several hours of air operations may be simulated in minutes, and traffic throughput, runway utilization, and other performance measures summarized immediately. Many different scenarios can be tested simply by changing model parameters: traffic load and patterns may be adjusted, and holding patterns may be relocated by moving navigational beacons or changing aircraft routes. It is conceivable that as many fast-time simulations could be run in one day as in a month of real- time simulation.

Despite these advantages, fast-time simulation cannot replace real-time simulation. Rather, the two methods are complementary: the feasibility or efficiency of a number of options and alternative scenarios may be tested in fast time, and the most promising may then be run in real time for complete evaluation and pre-implementation training. The availability of both methods allows their cost-effective application in combination.

1.3 The Terminal-Area Fast-Time Simulation Model

In preparation for the design of a production fast-time simulation system for the R&E Centre, the prototype fast-time model described in this thesis has been developed at the Centre. The model is intended to serve as a planning tool that will satisfy some of the current requirements for fast-time simulation, and also as a test-bed for fast-time algorithms and applications. The software has been written in the SIMULA language for the DECsystem-10 at the Research and Experimentation Centre, and is referred to as the TFS ("Terminal-Area Fast-Time Simulation") model. The model uses discrete and continuous simulation techniques to model the flow of arrival and departure traffic in the controlled airspace around an airport (a "terminal area"): continuous components of the model approximate the vertical and horizontal movements of aircraft in accordance with aircraft performance data; discrete components include an ATC process that maintains separation among aircraft to create a realistic traffic flow pattern.

The geographical configuration of the airspace, including airways, radio navigation aids, and runway locations, as well as the ATC policies to be used in controlling traffic in the airspace, may all be defined in data tables to create a simulation scenario. The traffic arrival and departure rates and probability distribution may be adjusted for each aircraft type in order to investigate system performance under different traffic loads.

While a simulation is running, various performance measures are collected, including aircraft delays, runway utilization, and total traffic throughput, and the statistics may be displayed graphically. As well, "snapshots" of aircraft positions may be displayed on an airspace map, so that traffic flow may be investigated in detail.

The model provides the Research and Experimentation Centre with a unique technology for the evaluation of airspace structures and control procedures, and with an economical alternative to real-time simulation. To date, the model has been used in actual planning projects, including a cost/benefit study of radar installations, and as a tool in the preparation and validation of flight paths and aircraft performance estimates for a real-time simulation exercise. A number of extensive fast-time studies are planned in 1987, pending the implementation of several extensions to the TFS model.

In the chapters that follow, the model is described in detail. Following an introduction to the basic concepts of air navigation and air-traffic control in Chapter 2, and a review in Chapter 3 of simulation principles as they apply to air traffic, including a review of relevant existing models, Chapter 4 describes the architecture of the model and its support programs, Chapter 5 describes the setup and results of a typical fast-time simulation experiment using TFS, and Chapter 6 discusses the current status of the TFS model and future directions for its development.

2. AIR NAVIGATION AND AIR-TRAFFIC CONTROL

As background for the discussions in Chapters 3 and 4 of air-traffic simulation and of the TFS simulation model, this section presents a somewhat simplified view of Canadian airspace and of Air-Traffic Control (ATC) operations. The description emphasizes those aspects of the air navigation and ATC systems that are relevant to the intended application of the TFS model.

TFS models the operations of a (significant) subset of the air traffic in Canadian Domestic Airspace:

. aircraft operating under Instrument Flight Rules, flying to a controlled airport along designated airways, and executing an IFR approach to the landing runway; and

. IFR aircraft departing from a controlled terminal via a designated airway.

While this class of air traffic excludes much of Canada's aviation activity (all Visual Flight Rules (VFR) operations and flights in uncontrolled airspace, for example), it does include the high-volume operations at major airports that are the main focus of air traffic systems planning.

More detailed information may be found in Transport Canada publications, including the Instrument Procedures Manual [1], A.I.P. Canada [2], and the ATC Manual of Operations [3]. The information in this chapter is based on these references.

2.1 The Air Navigation System

2.1.1 Navigation Aids

The important features of an airspace - significant geographic points ("fixes"), airways, approach and departure paths, and holding patterns - are all defined in relation to ground-based radio navigation aids, or "navaids". For the purposes of this discussion, four types of navaid are of interest: the VHF Omni-Directional Range (VOR), Distance Measuring Equipment (DME), the Instrument Landing System (ILS), and the Non-Directional Beacon (NDB).

- VHF Omni-Directional Range: The signal from a VOR encodes directional information so that an airborne VOR receiver can display the magnetic bearing from the station to the aircraft (referred to as a "VOR radial"). Thus a VOR-equipped aircraft can fly a track along a specified radial to or from a station, and VORs are the basis of the airway network used by IFR traffic in controlled airspace; airways are generally made up of straight-line segments defined by VOR radials.

- Distance Measuring Equipment: A DME facility is usually co-located with a VOR, and provides a means for an aircraft DME receiver to determine its distance to the facility. Thus the combination of these navaids provides a suitably equipped aircraft with its "polar coordinates" relative to the VOR/DME. As well as determining aircraft position relative to the facility itself, the VOR/DME signals are used to define significant geographical points (points at which position reports are required, or

points from which approach tracks are defined, for example) as radial/distance pairs. An aircraft may also fly a "DME arc" relative to the facility, a manoeuvre in which the aircraft flies along an arc of constant DME distance; a DME arc may form part of an IFR approach.
(The DME function is often provided by a military Tactical Air Navigation (TACAN) system co-located with the VOR. In this case, the facility is referred to as a VORTAC, but is functionally identical to a VOR/DME when used by civilian aircraft.)

- Instrument Landing System: ILS systems are located at airports, and an individual ILS is associated with a particular runway. The ILS transmits focussed radio signals (the "localizer" and "glide path") that allow airborne receivers to determine the aircraft's horizontal position with respect to the extended runway centre-line and its vertical position with respect to the standard glide-path to the runway threshold. At major airports, standard ILS approaches are published for use by ILS-equipped aircraft.

- Non-directional Beacon: An NDB is a low-frequency radio beacon. An aircraft may be equipped with a direction-finding receiver that will display the current bearing of the NDB, and that will indicate when the aircraft is passing over the NDB. NDBs are used as marker beacons in ILS installations to indicate critical points in the ILS approach path, and as non-precision approach aids in certain types of IFR approaches.

2.1.2 Fixes

A "fix" is a geographic point that is significant in air navigation: it may be a point at which an aircraft must transmit a position report to ATC, it may define the intersection of two airways, it may form part of the definition of an airport approach or departure track, or it may define the location of a holding pattern. The point may be specified in a variety of ways:

- as the location of a navaid (usually an NDB or VOR),
- as the intersection of radials from two VOR facilities,
- as a distance along a VOR radial from the facility (a "radial/DME" definition), or
- simply as a latitude and longitude.

Fixes that are not defined by the location of a named navaid are assigned a five-letter name (as illustrated in the following section).

2.1.3 Airways

Enroute from one airport to another, an IFR aircraft flying in controlled airspace will usually follow a path that is made up of segments of designated airways. These airways are defined by reference to VOR navaids, as illustrated in Figure 2.1, which shows a section of an aeronautical chart for the Calgary-Edmonton area. An aircraft flying from Calgary to Edmonton along VOR airway 21 (or "Victor 21"), for example, will depart Calgary along the 354 radial of the Calgary VOR (whose identifier is YYC) to the fix denoted DELBR, located at

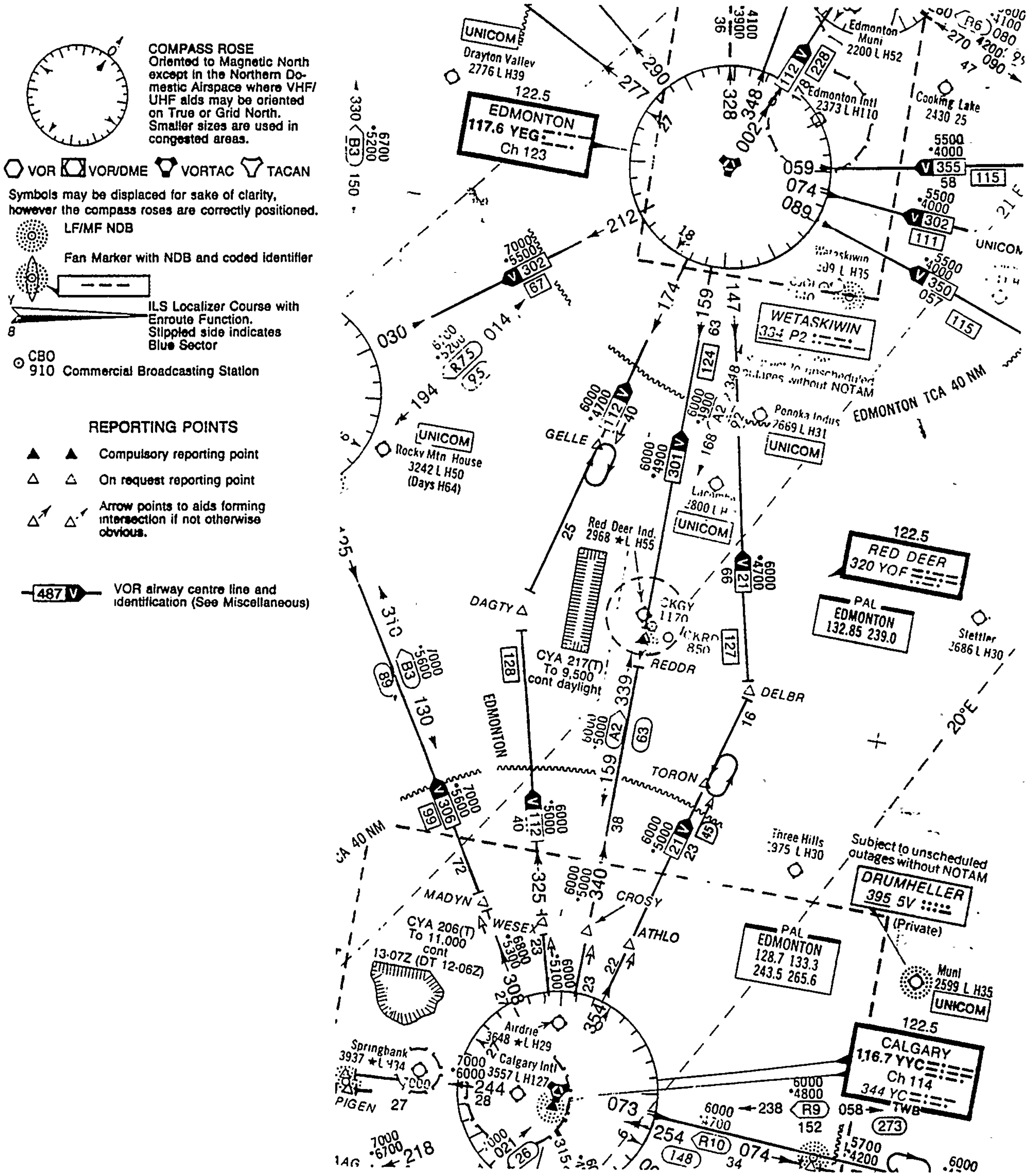


Figure 2.1: Section of Enroute Low-Altitude Chart for the Calgary-Edmonton Area

the intersection of the YYC 354 radial with the 147 radial of the Edmonton VOR (YEG), and thence via the YEG 147 radial toward Edmonton.

(Figure 2.1 is taken from an "Enroute Low Altitude" aeronautical chart, which depicts airways for use below 18,000' above sea level (ASL). The high-level airways depicted in "Enroute High Altitude" charts are defined in the same way by VORs, and so an example is not included here.)

2.1.4 Approaches

Once an IFR aircraft has reached the vicinity of its destination airport, it leaves its enroute airway, descends and reduces speed in a "transition phase" of its flight, and then executes an IFR approach to the airport, all in accordance with clearances from an ATC unit. The aircraft may simply be cleared by ATC to execute a sequence of manoeuvres that is published as a standard IFR approach to the airport, or, in busy terminal areas with radar surveillance, ATC may specify a sequence of headings ("radar vectors") and altitude changes that will bring the aircraft to a position from which the final phase of a standard approach may be carried out. (Radar control is discussed in more detail in Section 2.2.)

The basic objective of a standard IFR approach procedure may be stated as follows: using positional information provided by available navaids, the procedure defines a series of manoeuvres that will position the aircraft so that it is flying along the extended runway centre-line at an altitude that places it on the glide path to the runway threshold, at a suitable distance from the threshold, and at a suitable speed, for a landing to be executed. Obviously, the specific manoeuvres that are required to achieve this objective will depend on the navaids that are available; consequently, several different types of IFR approach are defined. For the purposes of this discussion, two types are of interest, the "straight- in" approach, and the "full", or "procedure turn" approach.

The straight-in approach is illustrated by Figure 2.2. The Initial Approach Fix (IAF) marks the departure of the aircraft from the enroute phase of its flight, and its entry to the initial approach phase, in which the aircraft flies to an Intermediate Fix (IF) from which the straight-in approach may be executed. Typically, the initial phase consists of a direct path, or perhaps a DME arc, to the IF. The IF generally lies on or close to the extended runway centre-line, and is defined as either a VOR/DME radial/distance, or as a distance along the ILS localizer. Once past the IF, the aircraft is in the intermediate phase of the approach, and is aligned with the runway; the subsequent approach track is defined by a VOR radial, in the case of a straight-in approach that is based on a VOR aligned with the runway, or by an ILS localizer, in the case of an ILS- based approach. The final approach phase begins, appropriately, at the Final Approach Fix (FAF), which is usually located about four nautical miles from the threshold. The FAF may be defined in various ways, but is usually an NDB, or an ILS "marker" beacon; the approach procedure definition will specify the precise altitude at which the FAF must be crossed.

The various forms of the full approach are illustrated

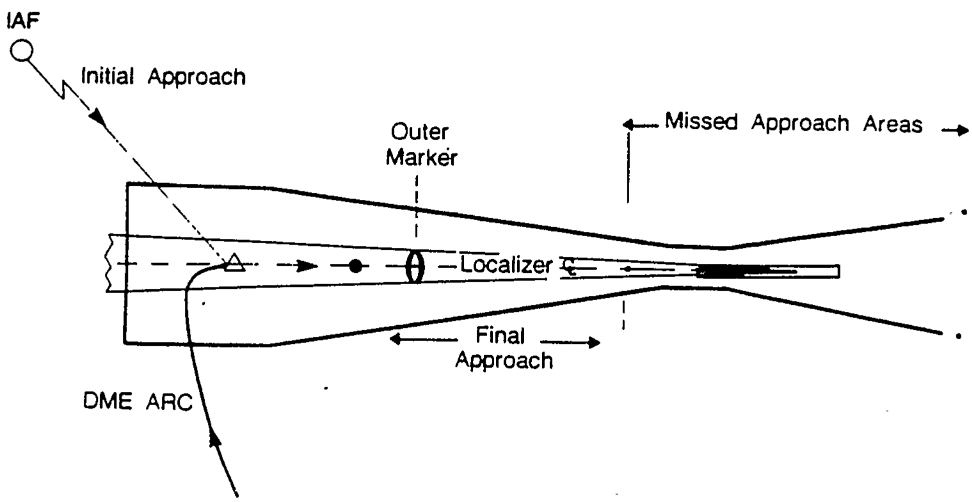


Figure 2.2: Straight-In Approach Geometry

by Figure 2.3. A procedure turn is required at the FAF when the navaids available to the aircraft are not precise enough or located appropriately to permit the accurate determination of position that is required during the transition to a straight-in approach. Typically, the execution of a procedure turn is based on an NDB: the arriving aircraft flies directly to the NDB, using a direction-finding receiver tuned to the beacon, and once its precise position is established at the time of crossing the NDB, begins the procedure turn as shown in Figure 2.3. From the initial known position above the NDB, a series of timed manoeuvres and a calculated descent rate will return the aircraft to the NDB (the FAF for this procedure) at a heading that aligns it with the runway and at an altitude that lies on the glide path.

For any IFR approach, a "decision height" is specified. At this height, if poor visibility prevents the pilot from establishing adequate visual ground reference, he must abort his approach and execute a missed-approach procedure, which usually involves flying clear of the airport area to a "missed approach fix" (MAF) before attempting another approach. (At the other extreme, in clear weather, if the pilot sights the runway when the aircraft is still some distance from the airport, with the agreement of ATC the IFR approach may be discontinued in favour of a shorter visual approach procedure, although this is not usual in heavy traffic at a busy airport.)

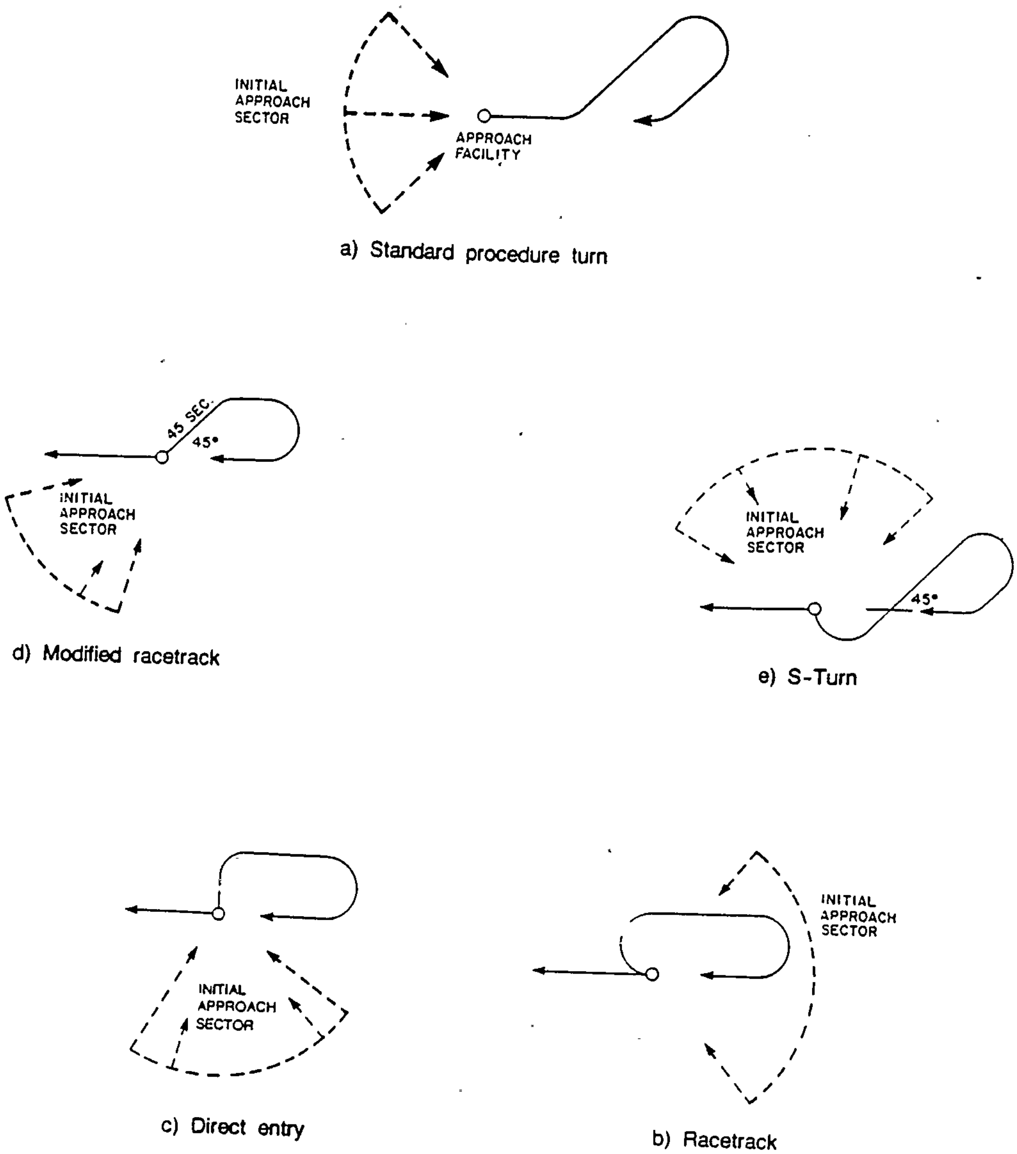


Figure 2.3: Procedure Turns

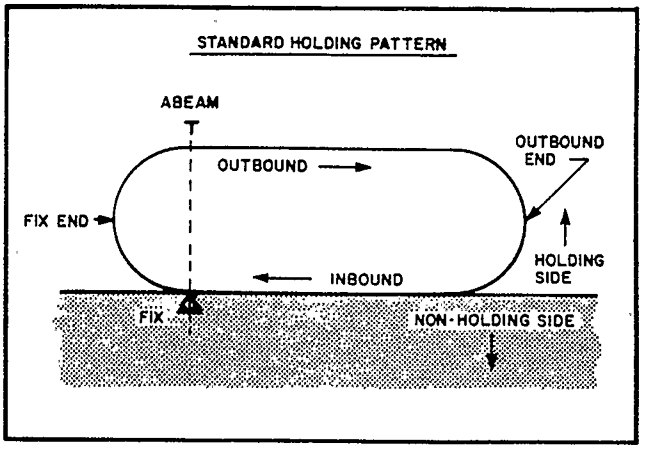
2.1.5 Holding Patterns

Under various circumstances, usually when traffic congestion at the destination airport makes it necessary to delay the approach of an arriving aircraft, the aircraft will be placed in a holding pattern by ATC. A holding pattern is defined in relation to a fix as shown in Figure 2.4; the heading to be flown on the inbound leg (toward the fix) is specified as part of the holding pattern definition, as is the rate of the turns ("standard rate" is three degrees per second) and the length of time to fly outbound (one minute or ninety seconds, depending on altitude) before turning to the inbound heading and returning to the fix to begin the manoeuvre again.

2.2 Air-Traffic Control

This section will briefly summarize the separation rules applied by air-traffic controllers in a controlled terminal area. (The airspace around busy airports is usually designated as a Terminal Control Area and controlled from a Terminal Control Unit. The control of enroute aircraft by Area Control Centres is not discussed here, as it lies outside the scope of the TFS model. The focus of the work reported here was the traffic flow in a terminal area, and it was simply - and realistically - assumed that arriving aircraft would be adequately separated by enroute ATC at the time they were "handed off" to the terminal ATC as they approached the boundary of the terminal airspace simulated in TFS.)

Two basic sets of separation standards are used for airport arrivals and departures: radar standards, and



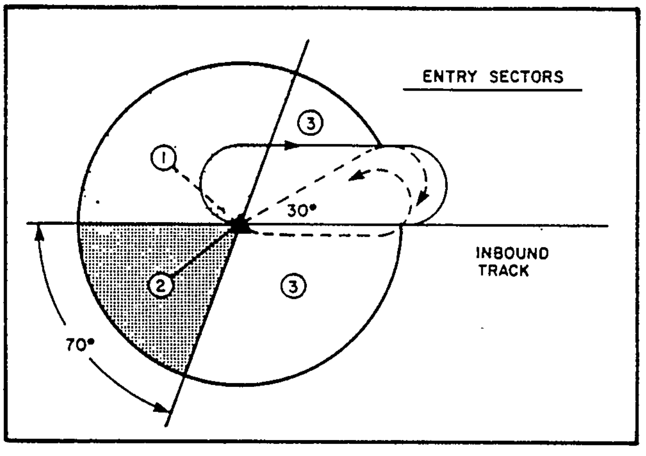


Figure 2.4: Holding Patterns

procedural standards. Radar separation standards are applied in terminal areas in which aircraft are under radar surveillance by ATC, as would be the case in Terminal Control Areas surrounding major airports. Because of the precise position information that is continuously available to radar controllers, less separation is needed among aircraft in radar-controlled areas than in non-radar control zones. Without radar surveillance, controllers must apply procedural standards based on radio position reports.

In a radar-controlled terminal area, an aircraft will be cleared to an approach path if it will remain horizontally or vertically separated from other aircraft as it makes the approach and lands. The usual horizontal and vertical minima are three nautical miles, and 1000 feet, respectively.

The approach procedures in a radar controlled terminal area may include radar vectoring of aircraft by ATC. Vectoring provides more flexibility in approach routing than is afforded by the set of standard approaches, and allows the controller to assign routes to the aircraft in the terminal area in a way that maximizes use of the available airspace, while observing the required separation standards. (Figure 4.5 shows typical vectoring paths in schematic form.) The use of vectoring allows the controller to add varying distances to the approach tracks of arriving aircraft as necessary, to ensure that they are separated horizontally in later phases of the approach when all the arrivals must be aligned along the glide path, and vertical separation is impossible.

In a non-radar terminal area, required vertical separation among arrivals is 1000 feet, and horizontal separation is provided by a combination of lateral separation standards defined in terms of protected air-space around airways and holding patterns, and longitudinal separation standards which specify time intervals that must separate successive aircraft passing over a fix. Longitudinal separation standards vary with the speed relationships of successive aircraft (a less stringent standard is applied when a slow aircraft follows a fast aircraft, for example).

The separation standards applied to departing aircraft are essentially the same, with the following additional rules:

- Departure clearances are given only when the next arrival is a specified time or distance from landing;
- Longitudinal separation of departures may be increased because of wake turbulence caused by departing heavy aircraft;
- Longitudinal separation of departures may be reduced when successive departures are proceeding on diverging tracks;
- Separation of arrivals and departures is often ensured by defining areas in which arrivals must fly above a certain altitude, and departures remain below the altitude;

This section has given a brief summary of some of the principles of air-traffic control as they apply to the control of the airspace near an airport. The details of ATC operations and separation standards may be found in the ATC Manual of Operations [3]. The implementation of ATC functions in the TFS model itself is discussed further in Chapter 4.

3. AIR-TRAFFIC MODELLING

Analytical modelling methods, fast-time simulation, and real-time simulation are all used as tools in air-traffic systems planning. As background to the description of the TFS fast-time model in following chapters, and to place TFS in the context of other related work, this chapter presents examples of planning questions that are suitable for the application of modelling methods, reviews the various types of models that have been developed, including software used by the Canadian Department of Transport (DOT) and the United States Federal Aviation Administration (FAA), and concludes with a more detailed description of the existing fast-time simulation models to which TFS is related.

3.1 Air-Traffic Applications of Modelling

The work of the DOT Research and Experimentation (R&E) Centre provides many examples of air-traffic planning problems to which modelling methods are applicable. For example, real-time simulation exercises have been conducted at the R&E Centre in order to choose a set of holding fixes around the Toronto Terminal Control Area from which aircraft may be "metered" into the terminal airspace at a rate that can be handled by the terminal controllers. The procedural feasibility of each proposed set of metering fixes was evaluated in terms of fuel efficiency, the amount of inter-controller communication required to coordinate the arrival flights, and the frequency of arrival conflicts with enroute and departure traffic. Similar exercises have been conducted for other Canadian airports to determine the optimum location of holding fixes, the location of departure "gates" on the terminal boundary, and the metering rate at which traffic can be accepted. (While the inclusion of real-time simulation in a discussion of "models" may stretch the term somewhat, these problems addressed at DOT using real-time simulation would be typical candidates for the application of analytical or fast-time modelling and illustrate the target problem domain for TFS. In the remainder of this thesis, "modelling" refers to analytical and fast-time methods.)

As another example of modelling requirements arising at DOT, the traffic volume at an uncontrolled airport had increased to the point where it would soon become necessary to upgrade the airport's navigation facilities and to institute some form of air-traffic control. The alternatives under consideration ranged from procedural control with various combinations of navaids (NDBs alone, or NDBs plus a VOR/DME), to full radar control. These alternatives were to be evaluated with respect to the traffic capacity that they would accommodate, under various assumptions about the future volume of air traffic arriving and departing along the airways serving the region.

In the US, the FAA frequently encounters planning problems that require the use of modelling. A report summarizing FAA modelling requirements [4] lists a large number of specific questions that have arisen from FAA planning requirements, and for which modelling was identified as the appropriate method of analysis. Table 3.1 presents a representative selection of these questions.

As the examples in the table demonstrate, typical FAA

What are the estimated capacity increases and delay reduction benefits at the busiest 30 airports of reducing longitudinal separation minima for non-heavy aircraft to 2.5 nautical miles? 2.0 miles?

What are the delay/fuel reduction benefits of removing the 250-knot speed restriction in terminal areas?

To what extent would a 1-mile in-trail reduction in departure spacing reduce delays at the Denver airport?

What would be the optimal size and location of holding areas for the relief of peak-period congestion at the Denver airport?

By how much would the strict segregation of arrivals and departures by speed and weight characteristics increase capacity at Atlanta airport?

What would the effect on capacity at JFK and LGA be if curved approaches were made possible by the installation of Microwave Landing Systems at those airports?

At JFK, how much would the operation of independent departure streams on runways 31L and 31R reduce delays?

At La Guardia, what would be the optimal location for a new high-speed exit from runway 13?

Table 3.1: Representative Problems to which Modelling was Applied by the FAA. From [A].

planning problems involve questions about the impact of new navigation equipment, airport runways, and ATC procedures on system performance measures such as traffic capacity, aircraft delays, fuel efficiency, and noise. It is clear that air- traffic models must represent the airspace and ATC procedures in considerable detail if they are to be useful in solving the very specific questions exemplified by Table 3.1.

The examples presented in this section are discussed briefly, but nevertheless give the flavour of typical problems of airspace organization and control procedures to which modelling methods are applicable, and which are the focus of the work reported here. It should also be noted for completeness that there are other, more specialized, areas of application for air- traffic modelling:

- Real-time, operational applications in which models provide short-term capacity and delay predictions as part of wide-area flight scheduling systems, or in which aircraft movements are modelled in order to provide forecasted flight paths to conflict-prediction systems [4,5];
- "Test-bed" applications in which air-traffic simulation provides positional data for testing the software components of ATC radar trackers or collision avoidance systems, for example [6];
- Models of aircraft motion in which the representation of aircraft dynamics is sufficiently detailed to model the effects of weather, navaid and avionics system performance, aircraft dynamic response, and pilot performance, on the temporal and

spatial deviations of an aircraft from the desired approach track profile. Such models may be used to estimate the attainable track accuracy for different combinations of equipment capabilities and weather, or to evaluate the benefits of alternative navigation aids [7,8,9].

Although these applications are outside the scope of the present work, they make use of simulation methodologies that are similar to those used in the application areas that are of interest here.

3.2 Approaches to Air-Traffic Modelling

There are two ways in which existing air-traffic models may be categorized: with respect to application area, and with respect to methodology. There are three application areas that are relevant here (again, other more specialized categories exist - see [4,10]):

- Runway capacity and delay models are used to estimate the rate of arrivals and departures that can be sustained for a given airport configuration, and to estimate the corresponding flight delays,
- Airport models are used to estimate aircraft delay and airport capacity for different taxiway, arrival gate, and runway configurations,
- Terminal airspace models are used to evaluate the performance of existing or hypothesized ATC systems in the airspace surrounding an airport.

The models applied in each of these areas may be classified with respect to methodology as analytical or fast-time [10]:

- Analytical models are usually based on queueing theory, and are relatively inexpensive to run on a computer. However, their range of application is limited by the difficulty of formulating tractable queueing problems from scenarios that include mixed traffic streams, complex route structures and sequencing rules, and time-varying traffic rates.

- Fast-time simulation methods provide more detailed information (such as individual aircraft flight histories, or estimates of delay for different aircraft types) and can be used to experiment with details of ATC procedure and airspace layout. Fast-time simulation of a system involves the development of computer programs that model selected aspects of the behaviour of the actual system. The states of the system and its components are represented by "state variables" in the model. In particular, time itself is a state variable, referred to as "simulation time", and is incremented in a stepwise fashion through the simulation period; as time advances, the values of other state variables are modified in accordance with transformational rules that represent selected aspects of the internal processes of the system. Simulation time may advance as quickly as the computer can calculate the required transformations and record the resulting state variables for later analysis - hence the term "fast-time" simulation.

The following sections review existing models in more detail, with particular emphasis on the relevant models of the US National Airspace System (NAS). The next section summarizes an MIT review of NAS models [10], amplifying the brief description of model types given above, and Section 3.2.2 is based on a recent MITRE Corporation report [4] that reviews FAA-owned models and discusses specific FAA requirements for modelling tools. These two sections complete the overview of air- traffic modelling approaches, and are followed by Section 3.3, which presents more specific information on existing fast-time models, as background to the discussion in Chapter 4 of the TFS fast-time model.

3.2.1 Classification of Air-Traffic Models

In a report to the FAA in 1979, Odoni and Simpson [10] reviewed analytical and simulation models of the National Airspace System developed since 1970. Although the discussion of individual models is somewhat dated, the general classification of model types and comments on their applicability are worth summarizing. In their review, Odoni and Simpson identified the following model categories:

1. Runway capacity and delay models represent arrival traffic flow in final approach and landing, as well as departure takeoff operations, for various runway configurations and traffic mixes.

   The capacity models are computationally simple, and use analytical and probabilistic methods to calculate estimates of hourly runway capacity, which is defined as the average number of movements (landings and takeoffs) that can occur on a given runway configuration under continuous demand. The adjustable parameters of the capacity models include separation standards, traffic mix, approach-track geometry, weather conditions, navigation systems in use, and pilot performance.

The delay models estimate aircraft delays due to runway congestion. The analytical delay models are most useful for the calculation of aggregate delay statistics, and are based on queueing theory for congested systems with time-varying demand and service rates; the computational algorithms involve the numerical solution of differential equations describing the system's behaviour. Simulation delay models are used when more detailed delay forecasts are required, such as the estimation of delays by aircraft type, or for estimating the effect of changes in the details of runway configuration.

2. Complete airport models represent "airside" operations at an airport, including aircraft movements on runways, taxiways, aprons, and gates. For a given airport configuration and specified arrival and departure rates, the models estimate aircraft delay and airport traffic capacity as determined by such factors as traffic congestion on taxiways, queueing for arrival gates, and runway occupancy times.

   The methodologies employed in these models range from highly detailed simulation models to simple analytical models. The simulation models provide information on the details of traffic flow, such as the utilization rate of individual taxiways, but the data-collection and computing costs of using these models may be very high; unless the detail is required, simpler analytical methods are more cost-effective.

3. <u>Terminal airspace models</u> represent operations such as holding, vectoring, sequencing, metering, and spacing in the terminal area. This category includes a diverse group of models designed primarily to evaluate the performance of air-traffic control systems in the terminal airspace. Performance may be defined variously in terms of capacity, delay, safety, and fuel efficiency.

   The category of terminal-area performance models may be further subdivided into "microscopic" models which represent aircraft motion in detailed systems of equations, and "macroscopic" models that represent aircraft as points traversing paths in three-dimensional terminal airspace in accordance with ATC separation rules and sequencing policies. The microscopic models are used in studies of control strategies that require precision manoeuvering, or in studies of new navigational and surveillance techniques, where the effects of aircraft dynamic response are significant. When the performance measures of interest depend on large-scale aspects of aircraft motion such as flying time between fixes, or the time required to reach an assigned altitude, the macroscopic models are more suitable.

   Whereas capacity and delay parameters may be estimated for runways and airports by relatively simple analytical models in some cases, simulation has proved to be the only feasible methodology for the modelling of terminal-area traffic flows. A useful model of terminal-area traffic must represent the operations of holding, vectoring, sequencing, metering, and spacing in a geographically complex terminal area. The microscopic terminal-area models in use at the time of the Odoni and Simpson review used continuous simulation methods, and the macroscopic models used discrete-event methods (which are "considerably more economical but the simplifying assumptions are occasionally more severe").

4. Air-route traffic models represent FAA enroute control centre (Air Route Traffic Control Center) operations, airway flows, airway intersections, enroute flow control, and the communications workload of controllers. Because enroute aircraft fly at constant altitude and speed for extended periods of time, the models in this class tend to be simpler than terminal-area models.

   Air-route capacity models are analytical models that compute the capacities of en route airway segments and of airway intersections; capacity is determined by estimating the frequency of overtaking and crossing conflicts as a function of traffic load and composition, spacing rules, and airway geometry.

   Models of ATC communications apply queueing theory or simulation methods to the analysis of voice-communication traffic between pilot and controller.

5. Models of major NAS segments represent departure, enroute and arrival traffic movements within a group of airports and enroute sectors and are applied to system-wide performance issues, such as the propagation of delays through the system. Because of

the difficulty of representing a large airspace at a useful level of detail, very few of these large-scale models existed at the time of the Odoni and Simpson report. All of the existing models were simulation models.

In addition to the five categories of model described above, the report identifies three other model types which have limited relevance to the TFS project: <u>controller workload and performance models</u> that represent the actions of enroute controllers under various operating conditions; <u>safety-related models</u> which compute collision probabilities, deviations from prescribed flight paths, and other safety-related measures; and <u>noise-related models</u> that estimate aircraft noise levels in airport areas.

Two principal conclusions may be drawn from this MIT review of models of the US National Airspace System:

- Models have been developed for a wide range of NAS components, including detailed "microscopic" models of aircraft movements on final approach, and on airport runways and taxiways, to models of large sections of the NAS. Many of these models are in use operationally, and are judged to be successful applications of modelling to systems planning.

- The "best" modelling methodology depends on the purpose and scope of the model: simple analytical models are accurate and cost-effective for the estimation of aggregate, or average, system performance for a relatively simple representation of the modelled airspace, whereas the complexities of aircraft movement in the terminal and enroute airspace require the application of computer simulation methods.

These conclusions are reinforced by the review of FAA-owned models in the next section.

3.2.2 FAA Models

In 1985, the FAA published a study by Barrer and Weiss (of the MITRE Corporation) of the agency's requirements for capacity, delay, noise, and fuel models. Some of the specific modelling problems identified by the study have been presented in Section 3.1 above, and the more general requirements are briefly summarized here. The report also includes a review of airspace models that is more recent than Odoni and Simpson, although restricted to models owned by the FAA. The models of interest in this report are those that predict the effect of air traffic control on airport capacity and delay, on noise and on fuel consumption.

FAA modelling requirements include :

- The analysis of capacity and delay: in planning, there is a need for models that predict change in capacity when navaid equipment or runway layout changes. The models are needed for the analysis of site-specific questions relating to very specific engineering and development plans. There are also real-time, operational requirements: in recent years, there has been a growing emphasis on system-wide scheduling and sequencing of flights, and models are used operationally to produce short-range predictions of capacity and delay.

- Analysis of noise patterns around airports for the analysis of the environmental impact of airport and terminal area changes.

- Analysis of fuel consumption: because an aircraft's flight profile determines its fuel consumption, ATC procedures are a major determinant of fuel efficiency, and there is a requirement to model the interaction of these factors.

Noise and fuel consumption models are not relevant to the TFS model and are not discussed further here. (Although a fuel-consumption mode is a natural extension to the TFS model; in fact the primary FAA fuel model is based on the SIMMOD fast-time delay model discussed below.)

The capacity models described by Barrer and Weiss are used by the FAA for the estimation of the capacity change that will result from changes in airport layout or procedures. Pure capacity models are relatively simple, and since they do not take account of delays, are not useful in optimising or regulating traffic flow. The FAA Capacity Model, for example, calculates the minimum sustainable inter-arrival time from the average runway occupancy times of aircraft, the average separation among aircraft, and the traffic mix. (Version 1 of the program dates from 1975 and consists of 3000 lines of Fortran code; a typical model run uses 15 CPU seconds on an IBM 4341.)

Delay models are of more interest in the context of the present work. Delay models predict the average delay per aircraft, and so may be used to evaluate the effect on operating costs of a change in the physical or procedural factors that affect traffic flow. They are also useful in regulating traffic, because they can predict an airport's acceptance rate as a function of the airport's nominal capacity and of randomness in arrival rates. Both analytical and simulation delay models are owned by the FAA.

DELAYS, an MIT Delay Model written in Fortran, uses time-dependent queueing equations to estimate expected aircraft delay as a function of time of day, of arrival and departure rates, of average arrival and departure service times, and of priority scheme. Because the system is described in such simple terms, many factors of operational interest are not represented explicitly in the model and thus cannot be investigated directly with the model.

Airfield Delay Simulation Model (ADSIM) is a fast-time, event-processing simulation written in Fortran in the early 1970s. ADSIM models the movement of aircraft in a very detailed representation of airport runway and taxi-way system, and reports traffic rates, travel times, and delays on the airport surface. Inputs to the model include airfield geography, ATC procedures, and aircraft runway occupancy times, taxi speeds, and gate service times. ADSIM is useful in investigating very detailed "what if" questions, but at a corresponding cost in data preparation, model calibration, and computer time.

The Airport and Airspace Delay and Fuel Consumption Model (SIMMOD) is a fast-time event processing model written in SIMSCRIPT II.5. It contains the Airport and Airspace Delay Model (AADM), an earlier Stanford Research Institute model. (This model is incorporated in SIMMOD as its delay component. It is discussed in more detail in Section 3.3 because of its relevance to the TFS model; at the time of the Barrer and Weiss report, AADM was the only version of SIMMOD to have been extensively evaluated).

SIMMOD represents the movement of individual aircraft through a node-link representation of the airways, the approach and departure tracks, and the airport surfaces of terminal and enroute airspace. Air traffic control procedures, including application of separation standards, vectoring, and speed control, are simulated. The model estimates the effect on delay and fuel consumption of many parameters, including weather conditions, control strategies, separation standards, aircraft performance characteristics, airspace sectorization, and interaction among multiple airports. In searching for other models with which SIMMOD may be compared, Barrer and Weiss conclude that no other models provide all of the functions of SIMMOD. However, they compare SIMMOD's AADM delay sub-model, which is of most interest in this study, to a terminal-area delay model developed for NASA by The Aerospace Corporation. The NASA model is discussed with AADM in Section 3.3.

Barrer and Weiss conclude that there is still a real need for a general airspace model that can be used in planning arrival and departure routes in complex terminal areas. In their view, the fundamental challenge in airspace planning is the development of an airspace organization that will "achieve the multiple objectives of maximizing capacity and fuel efficiency while minimizing noise and delays". To support the planning process, an airspace model must estimate the effect on performance measures (capacity, delay, noise, fuel) of changes in factors such as aircraft traffic composition, ATC rules and procedures, airspace organization, and the location and capabilities of navaids and surveillance equipment. Of the currently available FAA models, Barrer and Weiss judge that SIMMOD is closest to meeting these requirements, but will fully satisfy them only after a substantial development effort.

3.3 Fast-Time Models

The AADM delay component of the SIMMOD model is described in a report to the FAA by Bobick and Couluris [11]. AADM was developed by SRI International for the FAA; it is an event-step simulation written in SIMSCRIPT II.5, and simulates the flight of aircraft through controlled enroute and terminal airspace, represented as a link and node route network. The AADM produces reports of aircraft delay and travelling time including average delay and time per aircraft for each route.

The logic of AADM is described in some detail here, for later comparison to the design of the TFS program.

The model's ATC logic includes Tactical control (maintenance of separation on individual inter-node links by vectoring, speed control, and assignment of aircraft to holding patterns), Sequencing control ("sequencing and spacing of aircraft for downstream merges" using the same methods as Tactical control) and multi-sector Strategic control (balancing traffic flows by controlling the rate of traffic entry at boundary nodes feeding into a sub-network that is under Tactical and Sequencing control):

- <u>Tactical control:</u> when an aircraft arrives at a node, Tactical control attempts to find a legal way for the aircraft to proceed to the next node. If successful, an "arrival" event is scheduled at the next node; otherwise the aircraft is placed on the current node's holding queue to wait for a re-try. In the search for a conflict-free path to the next node, a number of control actions may be used (speed changes, vectoring, path-stretching, and holding) to vary the aircraft's arrival time at the next node; the overall control strategy may be adjusted by the model user.

- <u>Sequencing control:</u> an optional level of control in which a set of source nodes from which traffic eventually converges on a downstream "post" node may be designated for Sequencing control, which attempts to minimize conflicts at the post node.

- <u>Strategic control:</u> an optional level of control in which separation rules at traffic entry points on the sector boundary are periodically adjusted to maintain acceptable traffic volumes in a sector; based on periodically updated forecasts of traffic rates at downstream post nodes.

One important component of AADM models the interface between airport and airspace: when an arrival event occurs at an airport node, the events associated with a landing (or missed approach) are scheduled , including an event that will set flags to block subsequent landings and takeoffs at the time when the arrival will be within a certain distance of touchdown. Whenever an event occurs that affects the permissibility of a departure, the departure queue is inspected. If a waiting flight can be cleared for departure, a departure event is scheduled, and a node "arrival" event is also scheduled for the time when the departure will lift off and enter the airspace.

Thus the AADM component of SIMMOD is a standard event-step simulation model. Aircraft motion is simulated only to the extent that the scheduling of node-arrival events is based on the calculation of inter-node flying times from aircraft performance data, weather parameters, and ATC control actions. The airspace and ATC representation is, however, very flexible, and the model has been used to model extremely complex airspaces.

The Aerospace Corporation [12] terminal-area simulation program was developed for NASA in the mid-1970s, and is designed to simulate the flow of high-density air traffic in the controlled airspace of a terminal area. The model is capable of simulating a terminal area that contains several airports, and has been used to model the operations of the three major New York airports. The authors of the model chose fast-time simulation methodology because they concluded that, consistent with the experience at FAA, analytical methods may be used successfully in modelling aircraft movements between the final approach fix and the runway, but are impractical for modelling the complex interactions of aircraft in the entire terminal area airspace.

Arrivals are generated by the model at some distance from the terminal area and proceed to their "feeder fixes", the geographic points from which approach tracks originate - each track defines a path linking the feeder fix to the final glide slope to the runway. For each arrival, a simulated air-traffic control function searches among the set of available approach tracks for one which satisfies the separation rules that have been specified for the simulation run. If the search is unsuccessful, the ATC function clears the aircraft to a holding pattern until traffic conditions permit an approach clearance to be issued. Departures are generated at the airport, and departure clearances are interleaved with arrivals so that departure-arrival and inter-departure separation standards are maintained.

Obviously, the key to the performance of the simulated terminal-area system is effectiveness of the ATC model's search for an approach path for arriving aircraft, since the resulting choice of a track, altitude profile, and speed profile determines an aircraft's approach time and fuel consumption. The candidate paths are defined for each feeder fix as a family of horizontal approach tracks, a family of altitude profiles, and a family of speed profiles. The set of candidate paths is the set of combinations of track, altitude profile, and speed profile. The candidates are considered in turn, beginning with the optimal path (the shortest track and most efficient descent profile), and proceeding to less desirable paths, until a path is found that will allow the aircraft to approach and land while maintaining separation from other traffic in the terminal airspace.

The ATC component of the program interacts with a flight simulation component that models the motion of an aircraft along its assigned track, incorporating the effects of pilot response-time errors, and inaccuracy of navigational equipment.

The model is implemented in a Fortran program using the "unit advance" approach, to use Franta's terminology [13], in which the simulation clock is advanced in fixed time steps. At each step aircraft positions are updated, and various conditions are tested to determine whether any significant events have occurred (for example, whether an aircraft has strayed far enough from its assigned path that a corrective command is required from ATC; or whether an aircraft has reached a reporting point or a point at which a manoeuvre is to start).

From a simulation run, the Aerospace Corporation model produces plots of vertical and horizontal projections of each aircraft's path, and detailed records of the simulated runway operations, including a log of arrivals and departures.

As fast-time simulation models of terminal area traffic flow, the AADM and Aerospace models have much in common with the TFS model that is the subject of this thesis. Although TFS was intended to satisfy many of the same requirements as the American models, the unique features of Canadian airspace required that a greater variety of scenarios be accommodated in the design. Furthermore, as the following chapter explains, a much different simulation methodology was needed for TFS, because of specific requirements for the animated display of traffic movements in the simulated airspace, and for future extensions of the TFS model to include the communication process, weather effects, and navigation error.

4. THE TERMINAL-AREA FAST-TIME SIMULATION MODEL

The definition of design goals for a fast-time simulation model at the Research and Experimentation Centre began with the identification of fundamental user requirements. It was clear that a generalized model was needed for application to a wide range of Canadian terminal-area planning problems, and that the model should be expandable to larger and more complex air-spaces than a single terminal area. It was intended that TFS include the capabilities of the fast-time models discussed in Chapter 3, as well as features that would meet the specific requirements of R&E Centre users for a graphic interface.

The design goals for TFS may be divided into two categories: immediate goals for the first version of the terminal-area model, and goals for later versions.

The immediate goals required the implementation of the following functions in the initial version:

- Estimation of terminal airspace capacity and aircraft delays for user-defined airspace configurations, traffic mixes, and air-traffic control policies; the initial version of the model is restricted to single-airport terminal areas, and to single-runway airport operations.

- Simulation of either radar or procedural control in a terminal airspace. This capability is particularly relevant in Canadian airspace, much of which is not under surveillance by ATC radar; because US airspace is completely covered by radar, the FAA models described above are applicable only in radar environments.

- Display of airspace features, with an animated display of aircraft movements. This is an essential requirement, because of the user's need to visualize traffic flows in order to judge the degree to which the simulated traffic patterns are realistic. The display also facilitates the interpretation of simulation results.

Later versions of TFS will implement additional functions, including:

- The simulation of flight paths that incorporate realistic amounts of pilot, controller, and navigational system errors. The model then may be applied to problems such as defining the dimensions of the airpsace that must be protected for safe execution of various procedures [7]; estimating the deviations of aircraft arrival times at a designated approach fix from their scheduled times, as a function of assumed distributions of technical and navigation errors; or, determining the accuracy required of navigational systems if aircraft are to meet given standards of accuracy for fix arrival times [8,9].

- The simulation of air-to-ground and inter-controller communication in order to model the effect of communication delays on air-traffic flow. This is another feature that is particularly relevant in low-density Canadian airspace, where communication between pilot and controller may be indirect, with messages relayed through a Flight Service Station.

In areas of high traffic density, problems of interest include communications channel capacity, and frequency of controller communications for different ATC procedures and airspace structures.

. The simulation of weather effects on traffic movements: development of a weather sub-model, and modification of the aircraft motion and ATC processes to include weather factors.

. Extended terminal-area ATC functions that will simulate the simultaneous use of several arrival and departure runways; the operations of a control area containing several airports; the re-assignment of runways during active terminal operations; and the use of "speed control" (aircraft speed adjustments requested by the controller) to maintain separation on approach tracks.

Alternative implementation strategies for TFS were evaluated against these design goals for a terminal-area model. As well, the long-term goal of developing an expanded airspace model was considered in the choice of implementation strategy: it is intended that the TFS model become the first component of a fast-time airspace model which will be capable of simulating traffic flow in composite airspaces consisting of several terminal areas and enroute sectors.

4.1 Implementation Approach

Because TFS was intended for application to planning problems that require very detailed modelling of airspace structures and ATC procedures, it was decided that TFS should take the form of a representational model that closely parallels the structure of an airspace-ATC system. This design strategy, in which the major components of the model correspond closely to the components of the modelled physical system, was also expected to facilitate a "building-block" approach to the later development of models for composite airspaces consisting of several terminal areas and enroute sectors.

While it is relatively easy to formulate a model design based on this representational concept (compared to, say, the more abstract SIMMOD concept of the air-traffic system as an event-driven network flow), a modelling methodology must be chosen with care if the conceptual model is to be implemented efficiently, and at a reasonable cost.

The "process-interaction" simulation methodology, as implemented in the SIMULA language [13], is most appropriate when a system of many processes operating in parallel must be modelled, and was the methodology chosen for TFS. Features of the SIMULA language also support the creation of a library of submodel definitions that will provide a programming environment for the development of models of composite airspaces.

Several of the defined requirements for TFS - an animated airspace display, the modelling of pilot and navigation errors and of weather effects, and the implementation of speed control - implied that a continuous simulation component was needed in TFS to model the motion of individual aircraft. TFS was therefore implemented using the DISCO extension to SIMULA [14], which provides continuous simulation facilities. Thus, while airways, approach tracks, and departure tracks are represented in TFS as a network of nodes and links as they are in the FAA's event-step SIMMOD program, TFS, in contrast to SIMMOD, does not model aircraft movements by simply scheduling node-arrival events at the end of flying times calculated for fixed inter-node tracks, but rather simulates the manoeuvres that the aircraft would execute in flying from node to node. Thus, the exact track followed by an aircraft between nodes will depend on the manoeuvering characteristics of the aircraft. While the current version of the TFS program does not simulate manoeuvering errors and navigation errors, this design approach will facilitate the addition of these factors.

The preceding paragraphs have briefly summarized the rationale for the methodology chosen for the implementation of TFS; the following sections review related technical concepts and terminology related to fast-time simulation; review the features required in implementation languages for discrete, continuous, and combined simulation; and describe the architecture of TFS.

4.1.1 Concepts and Terminology

As described in Chapter 3, a fast-time simulation model represents the state of the modelled system in a set of "state variables" which, as simulation time advances, are modified in accordance with a set of transformational rules that represent selected aspects of the internal processes of the system.

Most fast-time simulation models may be characterized as either "discrete" or "continuous", depending on the way in which state variables change value with time. (Some "combined" models have both discrete and continuous components; as explained below, TFS is an example of such a model.) The state variables of a discrete model change value only at specified time points referred to as "events". In the model of a single- server queue, for example, the state of the system - represented by the number of customers waiting for service and the availability of the server - changes only when a new customer arrives or a customer's service is complete. In a continuous model, state variables vary continuously with time. In a typical continuous modelling application, the energy transfers among the components of a heating system can be modelled by a set of differential equations that define the time derivative of each component's energy content. The system's behaviour over time is simulated by integrating the equations at small time steps over the simulation period.

The TFS air-traffic simulation has both discrete and continuous components: aircraft altitude, position, and velocity change continuously with time, whereas discrete changes in system state occur when a clearance is issued, when an aircraft reports reaching an altitude, a fix, or the limit of a clearance, or when a runway becomes occupied or available. This illustrates the characteristics of combined simulation that distinguish it from discrete simulation:

- The system state is not constant between discrete events: some state variables (aircraft positions, in this case) are continuously changing between successive discrete events.
- The occurrence of a discrete event may be triggered when a continuous variable reaches a threshold value

(in this case, when an aircraft reaches an assigned altitude, or passes a fix).

Finally, TFS, in common with most simulation models, is "stochastic" (or "probabilistic"), meaning that some model state variables take on values according to a probability distribution. System performance often depends primarily on the system's response to an unpredictable, fluctuating input. Customer waiting time in a service queue, for example, is sensitive to the occurrence of peaks and valleys in the rate of customer arrivals, and on the varying service times demanded by the sequence of customers. The average customer load may be well within the capabilities of the server, but if customer arrivals are unevenly distributed in time, unacceptably long queues may develop during traffic bursts. In the modelling of such a system, random number generators are used to calculate successive customer inter-arrival times and service times. Stochastic modelling is clearly appropriate in air-traffic simulation because of the random nature of much of the traffic flow, including random deviations in scheduled arrival and departure times.

4.1.2 Language Facilities for Discrete Simulation - SIMULA

In a program that implements a discrete simulation model, several general simulation functions must be performed:

(a) Time management:

- recording the passage of simulated time by maintaining a system clock that is advanced in discrete jumps from one event time to the next.

(b) Event synchronization and bookkeeping:

- facilities for keeping a list of pending events; the list typically consists of event notices giving the time and type of each pending event, and is updated frequently as new events are scheduled and previously scheduled events are re-scheduled.

- a control function to ensure that events occur in sequence and at the correct time.

(c) Queue management:

- facilities for creating queues and managing them according to various queue disciplines (an essential function, since most simulations involve competition for resources).

(d) Utility functions:

- generation of random variables from a variety of probability distributions.

- gathering of performance measures such as mean, maximum, and minimum queue lengths and waiting times.

- reporting of simulation results, including summaries of performance measures, and event traces.

These general discrete simulation functions are provided as standard facilities in a number of languages, including SIMULA [13,15,16], the language chosen for the development of TFS. SIMULA provides program statements for the scheduling of events, queue management, and performance reporting, as well as implicit control functions for time management and event synchronization. SIMULA implements the "process view" of simulation in which a simulation model may be formulated as a set of interacting, parallel processes; the behaviour of each system component is described by a SIMULA program as a process operating in parallel with the other components, interacting with them as required for synchronization and resource sharing. As noted earlier, this approach provides the flexibility required for the implementation of a detailed, representational model such as TFS.

SIMULA provides the discrete simulation functions described above in two classes: SIMSET and SIMULATION. (The "class" is a generalization of the Algol "block" and defines a package of data declarations, procedures, and class definitions, all of which are accessible in any program block prefixed by the class name. A programmer may also create extensions, or "subclasses", of a class "C" by writing the extensions in a new class definition prefixed by "C".)

Class SIMSET contains class definitions of list headers and list elements, as well as procedures for adding elements to lists, for deleting elements, and for performing other list manipulations. Subclasses of LINK, the list-element class, may have any set of attributes (data elements or procedures) definable in the SIMULA language, and thus may be used to represent any passive model entity or internal data structure. Objects of these subclasses may be created dynamically, and lists of the objects may be created and manipulated. These facilities of class SIMSET are used extensively in the simulation program for maintaining queues of messages, lists of aircraft, route tables, and queues of clearance requests.

Class SIMULATION defines a set of data types, procedures, and classes that provide the required discrete simulation features:

<u>class PROCESS:</u> Process components of the model may be represented as subclasses of PROCESS. Objects of this class may be created dynamically and may be placed on lists (PROCESS is itself a subclass of LINK), and may contain executable statements. The executable statements in process objects may execute as co-routines, and their execution may be suspended ("passivated") and reactivated to simulate the parallel operation of processes in the modelled system.

<u>Scheduling procedures:</u> Process co-routines may be suspended and activated at specified simulation clock times by built-in SIMULA procedures, which use an event list to coordinate the scheduling of the process objects. An executing process may, for example, call procedure HOLD to suspend its own execution for a specified time interval, a process may be PASSIVATEd until a condition is satisfied, and procedure ACTIVATE may be used to schedule a passivated process for execution.

<u>Random number generators:</u> Class SIMULATION defines a set of random number generation procedures that may be called from within the model program.

4.1.3 Language Facilities for Continuous Simulation

A number of basic required functions may also be identified for continuous simulation languages [17]. A continuous simulation language must provide a means of representing state variables and their derivatives. The model's continuous state variables must be identified to the language's integration routines, and at each integration step, the routines must have access to the definition of the state variable derivatives so that they may be calculated as many times as required by the particular numerical method that is being used.

These requirements are achieved in different ways by continuous simulation languages. In SLAM [18], for example, the user supplies a FORTRAN subroutine which may be called by the integration routines as required to calculate the values of state-variable derivatives. In ACSL [19], as another example, the user supplies a set of ACSL statements defining the way in which variables and derivatives are to be calculated, starting from specified initial conditions. These statements are translated by a pre-processor into a FORTRAN subroutine that is available to the integration procedures as required.

The approach taken in the SIMULA class DISCO [14], as the next section explains, exploits the features of the SIMULA language to provide a much more flexible means of defining the continuous components of a model. State variables and their derivatives are represented as attributes of dynamically created objects, and model equations are written as executable statements in a dynamically created process. Both variables and equations are activated and made available to the DISCO control routines by placing them on pre-defined lists.

4.1.4 Language Facilities for Combined Simulation - DISCO

Additional functions are required in simulation programs, such as the TFS model, that perform combined discrete-continuous simulation. Time management is complicated in combined simulation, because a sequence of integration steps must be fitted between successive discrete events, and the length of the last integration step in an interval must be adjusted to terminate at the discrete event time. Integration is interrupted for processing of discrete events, and then resumed until the time of the next discrete event.

Furthermore, in a combined model, several types of interactions between discrete and continuous components may occur, and functions for implementing these interactions are required:

- discrete events may modify the values of continuous state variables, or the values of parameters in differential equations.
- the structure of the differential equations representing a continuous process may be changed by a discrete event.
- the occurrence of certain discrete events (referred to as "state events") may be conditional on the value of a continuous state variable reaching a threshold.

DISCO [14] is a subclass of class SIMULATION that provides the additional facilities needed for combined discrete- continuous simulation:

<u>LINK class VARIABLE:</u> Objects of this class are used to represent continuously varying state variables in attributes STATE and RATE. For example, the altitude of an aircraft is represented in a VARIABLE object, with the current altitude given by STATE, and the current rate of descent or ascent by RATE.

<u>LINK class CONTINUOUS:</u> Objects of this class contain the executable statements that calculate the time-derivatives of all VARIABLE objects.

<u>procedure WAITUNTIL(B); name B; boolean B:</u> This procedure creates state events. The current (calling) process is passivated until B is true. B is evaluated at the end of each integration step, and so the activation of a discrete process may be made conditional on a boolean expression, which may involve continuous or discrete state variables.

In addition to these facilities which are used explicitly by the programmer, DISCO also contains procedures, invisible to the programmer, but written in standard SIMULA, which arrange for the execution of the statements in CONTINUOUS objects at the appropriate times, and the re-calculation of STATE and RATE for all active VARIABLE objects. These procedures also coordinate the timing of discrete events with the integration process, and ensure that state events occur when their conditions are satisfied.

In addition, DISCO includes definitions for several classes that may be used for data collection and reporting (these are based on the DEMOS [20] reporting functions).

The discussion of the Aircraft Subsystem in Section 4.2.1 describes how the functions provided in DISCO are used in TFS for solving the differential equations of the continuous components of the model, for coordinating the integration of these equations with the discrete components of the model, and for defining conditions for state events.

4.2 Architecture of the TFS Program

The SIMULA and DISCO processes that make up the TFS program are shown in Figure 4.1. Traffic-generator processes create aircraft and enter them in the simulated airspace. An aircraft-subsystem process controls the creation of aircraft and the simulation of aircraft motion, and reports the position and status of the aircraft to the ATC-subsystem process, which allocates altitudes and issues clearances. These component processes are controlled by data tables that define the geography of the airspace (fix locations, airways, approach tracks and departure tracks), the rate and locations at which aircraft appear in the system, aircraft performance characteristics, and ATC separation standards.

An object-oriented approach was adopted in designing the TFS simulation program, in order to take maximum advantage of SIMULA's capabilities for abstraction and information hiding. The most obvious example of object-oriented design in the TFS program is the set of PROCESS subclasses associated with an aircraft. One set is created for each active aircraft, with attributes of altitude, speed, heading, and location, and with defined operations of route extension and altitude change.

Other examples may be found in the ATC process. ATC's view of the airspace is defined in part by a set of data-objects, each of which embodies the status information for an airspace feature (a holding fix, a stack, or a runway), and which has operations for "acquiring" and "releasing" the feature, similar to the RES construct of DEMOS [20]. Also, within the ATC

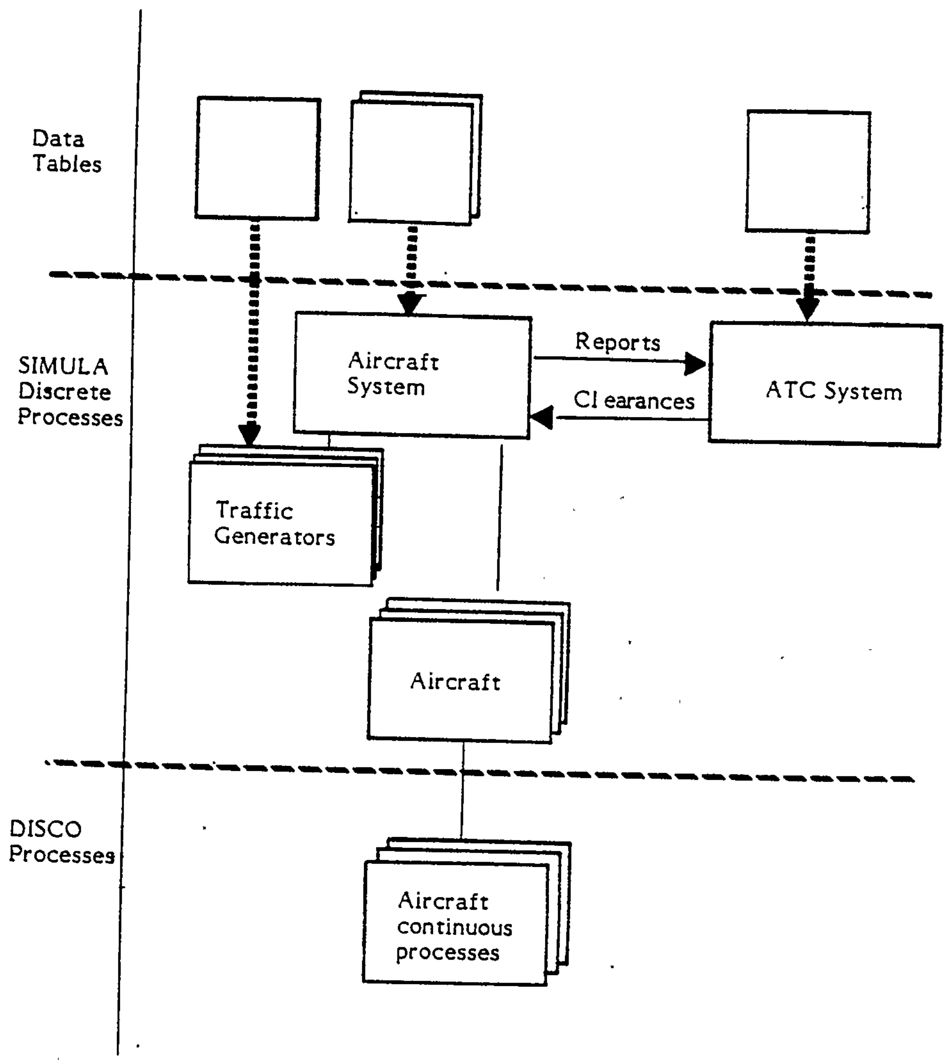


Figure 4.1 Overview of TFS Architecture

process, aircraft requests for clearance are mediated by process objects, one of which is created for controlling arrivals and one for controlling departures; each of these processes contains a table of forecasted aircraft positions and a set of procedures for operating on the table to perform separation checks and to produce clearance extensions and updated forecasts.

Consistent with the objective of producing a modular, building-block design, communication among the major processes of the TFS program is based on a message-passing model. The aircraft process-objects communicate with the air-traffic control process only by passing data structures analogous to messages, thus de-coupling these processes and allowing their internal processing to proceed asynchronously. Within these two main parts of the simulation program , modularity is achieved by allowing processes to communicate only by accessing each other's procedure attributes, in effect invoking one of the operations defined for the data type implemented in the target process.

The overview of TFS architecture given in Figure 4.1 is expanded in Figures 4.2 and 4.3, which show the major attributes of the component classes and the data flows among them. (The figures describe TFS using a graphical notation adapted from Buhr's [21] graphical design notation for Ada systems; Figure 4.4 defines the notation as used here.) The classes are also discussed below.

Three major PROCESS objects communicate by passing messages (actually "messages" are LINK objects - "sending a message" involves placing a message object on the receiver's message-queue list):

- the aircraft-subsystem object (AIRCRAFT_MODEL) controls the creation and simulated flight of the aircraft process objects; this process receives clearance messages from the ATC process and sends aircraft position reports to ATC. Some time before an aircraft actually enters the terminal area, an "advance notice" message is sent to ATC_MODEL so that it can take account of expected traffic as well as current traffic when assigning approach tracks.
- the ATC subsystem (ATC_MODEL) keeps track of the movements of simulated aircraft by monitoring position report messages received from the AIRCRAFT_MODEL, and directs aircraft movements by issuing clearance messages.
- A channel process (CHANNEL) through which messages between the aircraft and ATC processes pass. This is a skeleton process which was included with the intention of eventually simulating ATC communications, including message queueing, reaction time, delays and outages. At present, CHANNEL functions as a data-collection process. Since all messages pass through CHANNEL, the data required for all DISCO statistical objects is available there, and the statistics are updated from CHANNEL; message logs and other output files are also written from CHANNEL.

The following sections discuss the aircraft and ATC subsystems in more detail.

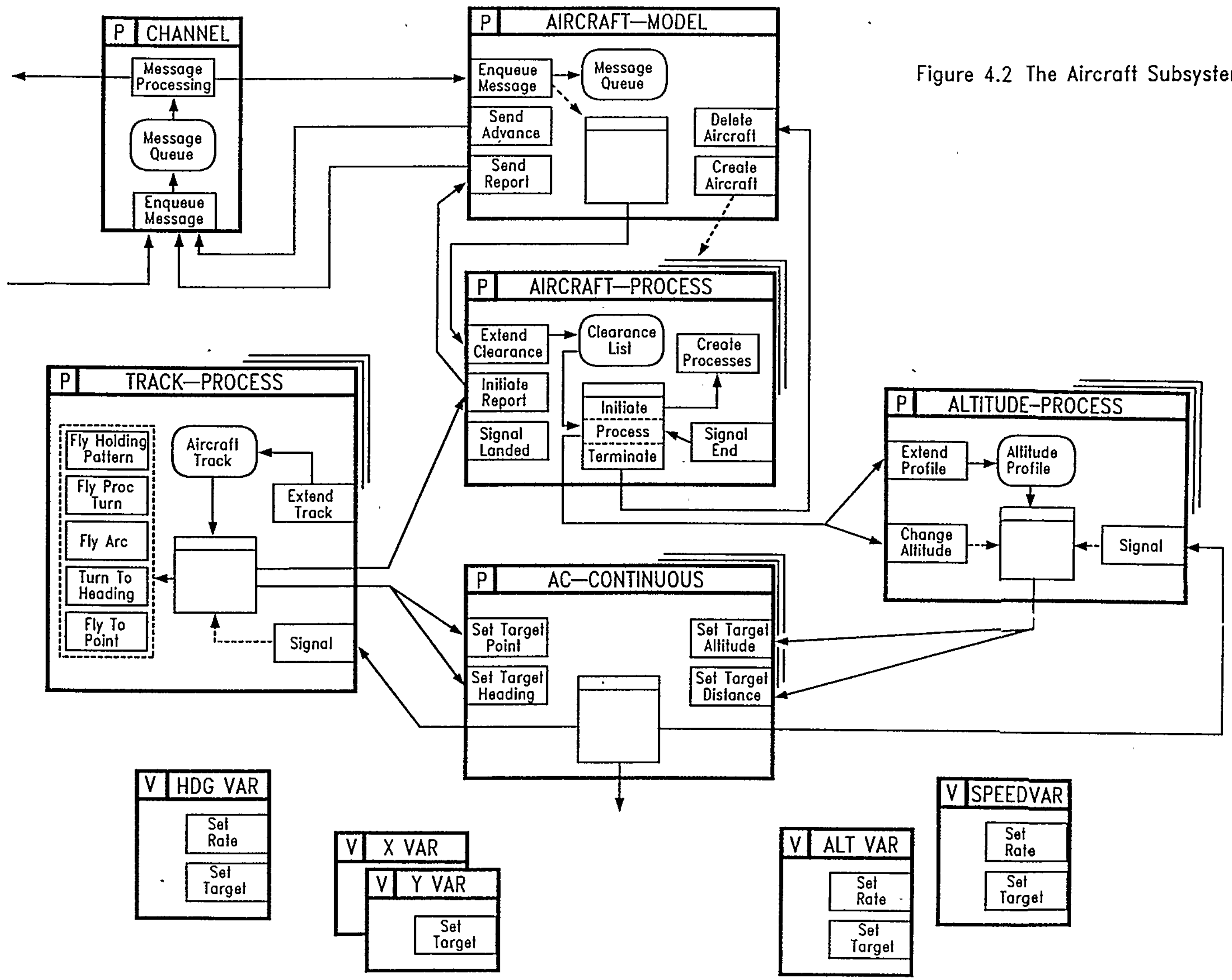


Figure 4.2 The Aircraft Subsystem

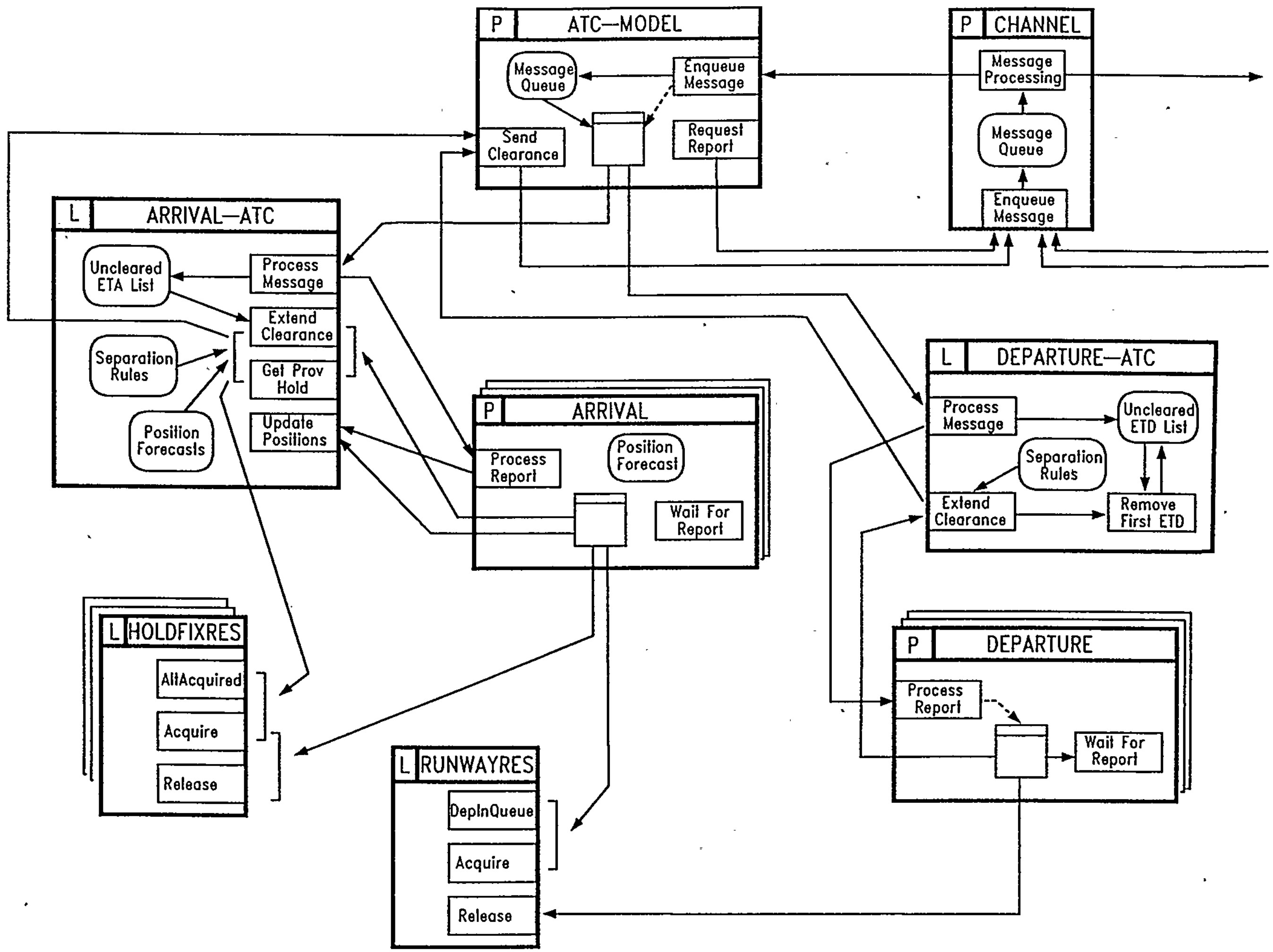
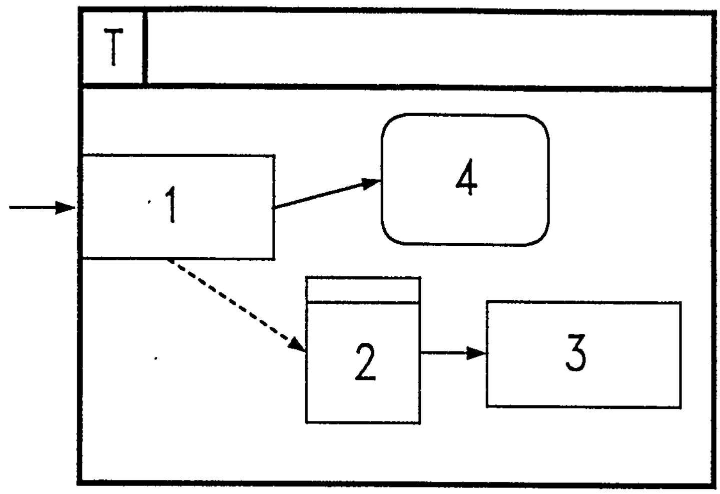


Figure 4.3 The ATC Subsystem



Denotes a SIMULA/DISCO object of type T, where:

T = P for Process
L for LINK
V for Variable
C for Continuous
H for Head

Box 1 denotes a procedure attribute called by other objects.

Box 2 denotes internal processing.

Box 3 denotes an internally used procedure.

Box 4 denotes a data structure.

Dashed lines indicate activation of a coroutine.

Solid lines indicate a flow of control or of data.

Figure 4.4 Object Notation

4.2.1 The Aircraft Subsystem

The aircraft subsystem process and its attribute procedures control the creation and simulated movement of air traffic by coordinating the activities of the traffic generators, aircraft-related discrete processes and aircraft-related continuous processes.

The traffic-generator process creates aircraft at the terminal-area entry points. The traffic stream is defined in one of two ways:

- as a file of individual flight plans, each of which specifies an aircraft type, a route for the aircraft, and an entry time; or

- by a set of distribution parameters for the random generation of inter-arrival times for each of several traffic streams; a "stream" is defined by a combination of aircraft type, route, and arrival rate (note that "arrival" is used in the general queueing sense here - "inter-arrival" times are defined for streams of both departures and arrivals).

When an aircraft is due to enter the terminal area, the traffic generator calls the CreateAircraft procedure of the AIRCRAFT_MODEL, passing the type code and flight plan of the aircraft to be created. The CreateAircraft procedure creates one PROCESS object of each of the following types:

PROCESS class AIRCRAFT_PROCESS: receives clearances from the ATC process (via the CHANNEL), and decodes them into directives for vertical and horizontal motion; also generates a position report whenever the aircraft reaches a point on the route which is a specified reporting point, or when ATC requests a report.

PROCESS class ALTITUDE_PROCESS: maintains a list representation of the aircraft's vertical profile along its planned route, and coordinates the continuous altitude-change process.

PROCESS class TRACK_PROCESS: maintains a list representation of the aircraft's track, and coordinates the continuous horizontal motion of the aircraft (including the manoeuvres that are executed to fly holding patterns, procedure turns, and final approaches).

Horizontal and vertical motion are simulated by separate processes so that discrete events and state events may be scheduled independently for altitude changes and manoeuvres, which may often be under way at the same time. State events may be sheduled to occur when a target altitude or heading is reached, and discrete events may be scheduled (using HOLD) at the completion of a timed manoeuvre.

When a new AIRCRAFT_PROCESS object is created, AIRCRAFT_MODEL places the new object on the active aircraft list, and thereafter controls its motion by passing clearance messages from ATC_MODEL to the AIRCRAFT_PROCESS ExtendClearanceList procedure. The AIRCRAFT_PROCESS in turn decodes the clearances into detailed directives to the associated altitude and track processes. Altitude changes are controlled by passing an altitude profile (specifying altitude as a function of distance travelled) to the ExtendProfile procedure of the associated ALTITUDE_PROCESS, or by calling the process's ChangeAltitude procedure, which directly initiates an altitude change. The track-segment list in the associated TRACK_PROCESS is extended by passing new segments to the process's ExtendTrack procedure. ATC_MODEL is advised of the aircraft's progress along its flight plan by AIRCRAFT_MODEL, which accesses the current status and location attributes of an aircraft and sends position reports to ATC_MODEL whenever an aircraft state event occurs (when an aircraft reaches a reporting point on its route or reaches an assigned altitude).

Five DISCO VARIABLE objects - speed, heading, X-coordinate, Y-coordinate,and altitude - are associated with each aircraft PROCESS object. An associated CONTINUOUS object calculates the time-derivatives of the VARIABLES at each integration time-step:

- the derivative of the speed VARIABLE is determined by interpolating in the aircraft performance tables to find the required speed for the aircraft's current altitude and flight status (for each aircraft type, functions relating speed to altitude are defined for the transition, manoeuvering, and departure flight phases); if the current speed is higher or lower than the required speed, deceleration or acceleration values are taken from the aircraft performance table and assigned to RATE.
- the derivative of heading depends simply on rate-of-turn, which is positive if the aircraft is turning right, negative if the aircraft is turning left, and zero otherwise. In the current version of TFS, turns are all made at "standard rate" (3 degrees per

second) although the program will easily accomodate different rates.

- the derivatives of the X and Y coordinates are calculated from speed and heading;

- the derivative of altitude is taken from the aircraft performance table for the aircraft type, and is a function of current altitude and flight phase (rate of ascent or descent may also be limited by various constraints - for example, the aircraft may be required by a clearance message to reach a particular altitude at a given location or time).

The VARIABLE objects in TFS are actually instances of a subclass ACVARIABLE, which has several specialized attributes in addition to the STATE and RATE attributes:

Target: the target STATE for the variable if the rate is non-zero (for example, the altitude to which an aircraft is climbing, or the coordinate of a fix to which the aircraft is flying);

Tolerance: the error-tolerance value for matching STATE to Target;

TargetDistance: the Target STATE must be reached by the time the aircraft has travelled this distance;

Similarly, the CONTINUOUS object associated with an aircraft process is actually an instance of a subclass ACCONTINUOUS. In addition to defining the calculation of the RATE values for each ACVARIABLE at each integration step, ACCONTINUOUS defines procedures for assigning values to the ACVARIABLE's tolerance and target attributes. ACCONTINUOUS also monitors the STATEs of changing variables to determine the point at which their targets are reached, and to initiate processing of state events. This last function involves the DISCO WAITUNTIL procedure. For example, when the AIRCRAFT_PROCESS receives a clearance extension that requires an altitude change, the following steps take place: the new altitude is passed to the ALTITUDE_PROCESS, which calls the SetTargetAltitude procedure of the aircraft's ACCONTINUOUS object, sets the altitude RATE appropriately, and then calls WAITUNTIL(SignalFlag). SignalFlag is an AIRCRAFT_PROCESS variable that is set to true when ACCONTINUOUS detects that the aircraft is within tolerance of the target altitude and calls the ALTITUDE_PROCESS Signal procedure; the ALTITUDE_PROCESS resumes execution, stops the altitude change, and continues processing altitude change requests if any are pending, or passivates until another request is received.

The processing involved in executing a heading or speed change is very similar to the altitude change processing described above. The case of the coordinate variables is somewhat different, since they change continually while the aircraft is active in the system and unlike the other variables, the coordinate variables do not go through the cycle of starting a change, reaching a target value, and stopping the change. An aircraft's path is made up of a sequence of geographic points joined by paths of various types (arcs, direct paths, vectoring areas), and each point is in effect a target point at which a variety of events may be triggered, including the termination of the manoeuvres required to fly the path to the point, and initiation of the manoeuvres for the next path segment. Position reports to ATC may also be generated when the point is a reporting point. The mechanism for triggering these events is similar to that for the other variables: when TRACK_PROCESS needs to wait until some manouevre is completed (until the aircraft is established on a new heading, or until a fix is reached, for example), the process will WAITUNTIL a flag is set by ACCONTINUOUS when the manoeuvre is completed. The TRACK_PROCESS will then initiate the next path segment or the next phase of a more complex manoeuvre (such as a procedure turn or holding pattern).

4.2.2 The ATC Subsystem and Associated Classes

The ATC function is modelled in TFS by three PROCESS classes and five LINK classes:

PROCESS class ATC_MODEL: one instance of this process is created and coordinates the ATC function, receiving position reports from the AIRCRAFT_MODEL and its processes, and passing back clearances.

PROCESS class ARRIVAL: one instance of this process is created for each arrival aircraft. It controls the sequence of searching for and issuing clearances and is activated each time the aircraft requires a further clearance; it provides a "queueable" entity to represent the aircraft's clearance requests.

PROCESS class DEPARTURE: one instance is created for each departing aircraft; its function is analogous to that of the ARRIVAL process.

LINK class ARRIVAL_ATC: one instance is created to hold status information for all arrivals, including flight plan, current location and forecasted positions; to provide functions that maintain the table of forecasted positions for all aircraft that have received approach clearances; and to provide a function that finds an approach clearance for an arrival by searching among the available approach tracks for a track that the aircraft may follow and remain separated from other aircraft. Subclass RDR_ARRIVAL_ATC is specialized for radar-controlled terminal areas, and NR_ARRIVAL_ATC is specialized for non-radar terminals.

LINK class DEPARTURE_ATC: maintains a list of all departures awaiting take-off clearances, and of all departed aircraft still in the terminal airspace.

LINK class HOLD_FIX_RES: one instance is created by the ATC_MODEL process for each holding fix in the terminal airspace, to coordinate the assignment of holding altitudes at the fix. Ownership of each available altitude is recorded in a table within the HOLD_FIX_RES. The table is updated by ExtendClearance using the HOLD_FIX_RES Acquire and Release procedures, which ensure that only one aircraft is assigned to each altitude.

LINK class RUNWAY_RES: represents the runway as a resource, to ensure that only one aircraft can have access to the runway at one time. An arrival must be able to acquire the runway once it is within a certain time of landing or execute a missed approach, and an aircraft awaiting departure must acquire the runway before taking off; arrivals own the runway until they have pulled off on a taxiway, and departures own the runway until takeoff.

The following paragraphs summarize the operation of these LINKS and PROCESSes in the simulation of arrival ATC functions; the departure ATC simulation is discussed at the end of the section.

When an "advance" message for a new arrival is received by the ATC_MODEL process from the AIRCRAFT_MODEL process, the ATC_MODEL passes the message to the ProcessMessage procedure of the ARRIVAL_ATC LINK, which then creates an ARRIVAL process. When the arrival enters the terminal airspace and sends a "report" message to ATC, the ARRIVAL process is activated and begins to coordinate the finding and issuing of the clearances required to land the aircraft.

If the aircraft has the earliest estimated landing time among the arrivals that are awaiting approach clearance, it is given first priority for landing clearance, and the aircraft ARRIVAL process calls the ExtendClearance procedure of the ARRIVAL_ATC process to search for a landing clearance. If the aircraft does not have the earliest estimate, the ARRIVAL process passivates until earlier aircraft receive their clearances.

The ExtendClearance procedure considers each possible approach track for the aircraft (there may be several, if the track from the aircraft's Initial Approach Fix to the runway has been defined to include a vectoring area). The procedure forecasts the aircraft's positions along the track at small time intervals, and calls ARRIVAL_ATC's SeparationCheck procedure to test each forecasted position for separation from the ARRIVAL_ATC table of the forecasted positions of arrivals that have already received approach clearance. The ExtendClearance procedure initially assumes that the aircraft's altitude profile along the tracks is a steady descent to the Final Approach Fix, but if necessary, the procedure will also check for separation along other profiles (a delayed, steeper descent, for example). If ExtendClearance is unable to find a feasible approach clearance, a free holding altitude is acquired from the IAF's HOLD_FIX_RES, the aircraft is cleared to a holding pattern at its IAF, and the ARRIVAL process begins a cycle of repeated attempts to find a clearance, separated by short delays (using SIMULA's hold procedure). Otherwise, an approach clearance is issued to the aircraft via the CHANNEL, and the aircraft's forecasted positions are passed to the UpdatePositions procedure of the ARRIVAL_ATC for entry in the forecasts table.

After the aircraft has been cleared to start its approach, the ARRIVAL process no longer has much work to do. The process passivates until the aircraft is within a certain distance of the runway, and then calls the Acquire procedure of the RUNWAY_RES. The process passivates until the aircraft reports leaving the runway; the runway is then released and the process terminates.

The simulation of departure ATC is similar to the arrival ATC simulation, but is simpler. The aircraft with the earliest estimated departure time is given priority for departure clearance; the DEPARTURE_ATC ExtendClearance procedure checks this aircraft's departure route (determined by the airway on which he is leaving the terminal area, and the departure tracks defined between the runway and the airway) to determine whether departure separation standards will be maintained. If not, the procedure estimates the delay required to establish separation from previous departures, and passivates the DEPARTURE process for that interval before re-trying. If separation would be maintained, the ExtendClearance procedure calls the RUNWAY_RES Acquire procedure; as long as the runway is owned by an arrival, or an arrival is queued for the runway, the DEPARTURE process is passivated in the runway queue.

4.3 The TFS Input Database

The geography of the terminal area, the traffic stream definitions, the aircraft performance characteristics, and the ATC separation standards that define a simulation scenario, are all entered on an IBM PC AT using a large dBASE III application program. Definition of a scenario involves the execution of a menu-driven sequence of data entry screens in which part or all of an existing simulation scenario may be copied and edited, or a new scenario created from scratch.

The following files of geographic information must be created for a scenario:

1. Fix Table: A fix definition consists of a fix name and a location; the location may be specified as a latitude/longitude, as a VOR name, as an NDB name, as a radial/DME, or as a radial intersection).

2. Airway Table: An airway definition consists of an airway name and a sequence of fix names.

3. Runway Table: A runway definition consists of the airport and runway names, the latitude and longitude of the runway threshold, the altitude of the runway

threshold, the name of the FAF for the runway, and the name of the Missed Approach Fix for the runway.

4. Initial Approach Fix Table: An IAF definition consists of a fix name and the name of the associated arrival airway; all approach tracks defined for the airway originate at the IAF.

5. Approach Track Table: An approach track consists of a track identifier, an airport name, a runway number, and a sequence of point and track segment definitions. The first point is the IAF for an airway, and the last point is the FAF for the runway. A track segment may be a direct path between two points, a DME arc between two points, or a VOR radial between two points. In addition, radar approaches can have "vectoring area" segments which may be used to simulate path stretching by radar vectoring. Figure 4.5 illustrates two types of vectoring area, showing the alternative paths from which ATC may choose in issuing radar approach clearances. Definitions of non-radar approach tracks include an approach-type designator that specifies a full or straight-in approach.

6. Departure Track Table: A departure track definition consists of a track identifier, an airport name, a runway number, the exit airways fed by the track, and a sequence of track-segment and point definitions.

The following <u>separation parameters</u> must be specified in the scenario definition: horizontal and vertical separation standards for radar and non-radar control, initial interdeparture vertical and horizontal

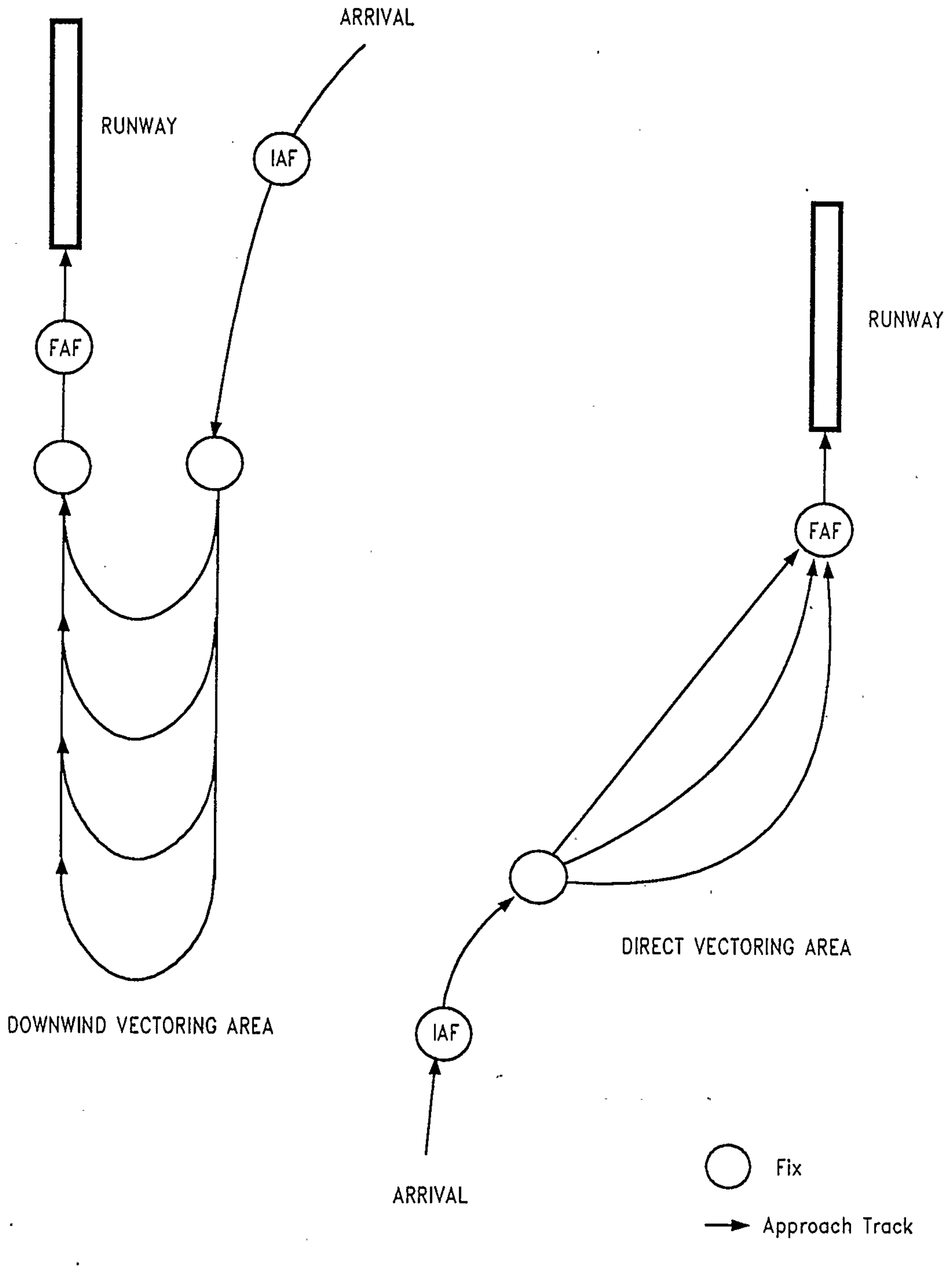


Figure 4.5 Vectoring Areas

separation (for both diverging and non-diverging tracks), minimum time and distance separation between a departure and the next arrival, and altitude restrictions for departures and arrivals.

The traffic for a scenario may be defined by a file of flight plans (each of which has an associated arrival or departure time) or by random traffic streams. A random traffic stream is defined by: an aircraft type identifier, an average inter-arrival time, departure and destination airport names, a route sequence of the form {fix name, airway name, fix name, airway name,...}, a range of entry altitudes and an entry descent rate for arrivals, and a variety of other parameters such as initial fuel load and random-number stream identifiers for each random parameter.

Aircraft performance characteristics are also input to TFS as part of a scenario definition. These characteristics include: final approach and landing speed; landing deceleration; take-off acceleration and speed; normal and maximum airborne acceleration, deceleration, and turning rates; and several characteristics that are defined in a table as functions of altitude, including enroute cruising, climbing, and descending speeds, climb rates for different fuel loads, and normal and maximum descent rates. Other aircraft data input to TFS include the runway occupancy table, which defines for each combination of aircraft type and runway identifier the minimum and maximum time that the aircraft may occupy the runway after landing.

Finally, the scenario definition is completed by a file that contains general parameters such as: the magnetic variation in the terminal area, a radar/non- radar flag, the arrival and departure runway names, the simulation start and end times, the number of replications of the scenario to be run, the seeds for the random number streams (each random parameter in the scenario definition has a stream identifier associated with it), and miscellaneous output switches.

Once the dBASE files have been prepared, they are transmitted to the DECsystem-10 and merged and verified by a pre-processor program, which calculates latitude and longitude for all points that have been defined in terms of VOR radials and DME distances. The resulting files are submitted to TFS.

4.4 TFS Output

As many as five output files may be produced for each replication of a TFS run. The Message Log File contains text versions of all messages sent between the AIRCRAFT_MODEL and ATC_MODEL processes. The Report File contains the summary statistics produced for the run by the DISCO statistical reporting facilities. The report gives COUNTs of arrivals, departures, missed approaches, and of aircraft that could not be given a clearance and were deleted (clearance failure is infrequent, but obviously significant - it is usually indicative of some inconsistency in the scenario definition). Other statistics describe the output of the individual random traffic streams and are used to verify the operation of the traffic generators.

The Aircraft Summary File contains a record for each aircraft that was created in the simulation run. The record summarizes the simulation history of the aircraft

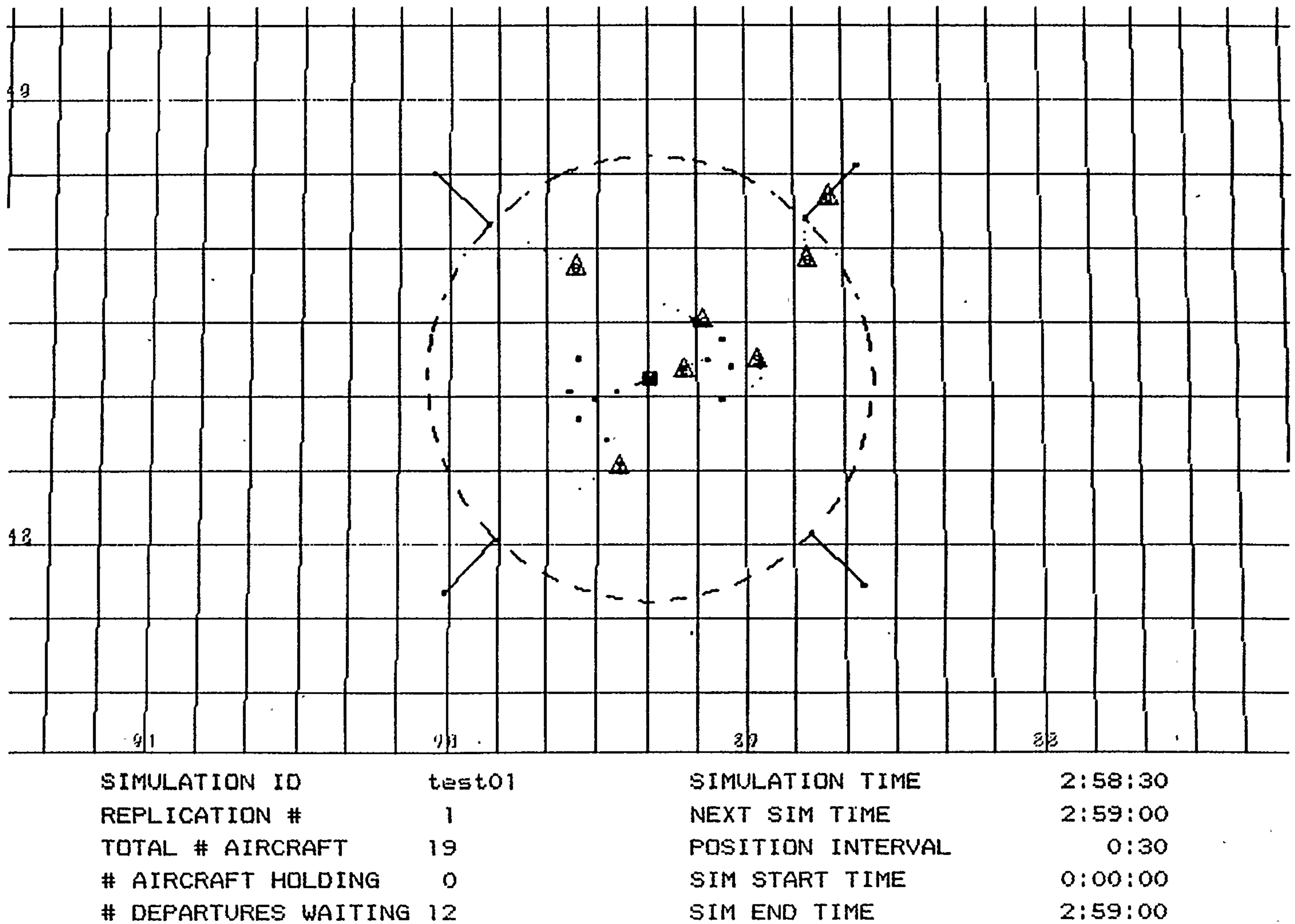
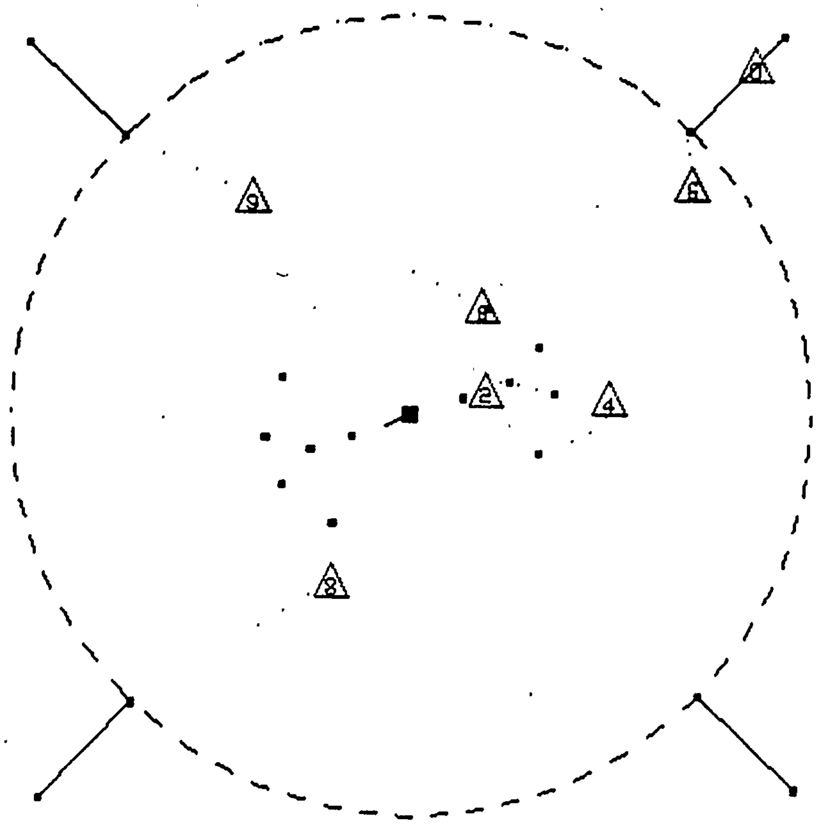


Figure 4.6 Airspace Display



SIMULATION ID	test01	SIMULATION TIME	2:58:00
REPLICATION #	1	NEXT SIM TIME	2:58:30
TOTAL # AIRCRAFT	19	POSITION INTERVAL	0:30
# AIRCRAFT HOLDING	0	SIM START TIME	0:00:00
# DEPARTURES WAITING	12	SIM END TIME	2:59:00

Figure 4.7 Zoomed Airspace Display

and includes the aircraft's creation time, landing time (for arrivals), terminal-area departure time (for departures), accumulated holding delays, and flying time along its approach or departure track.

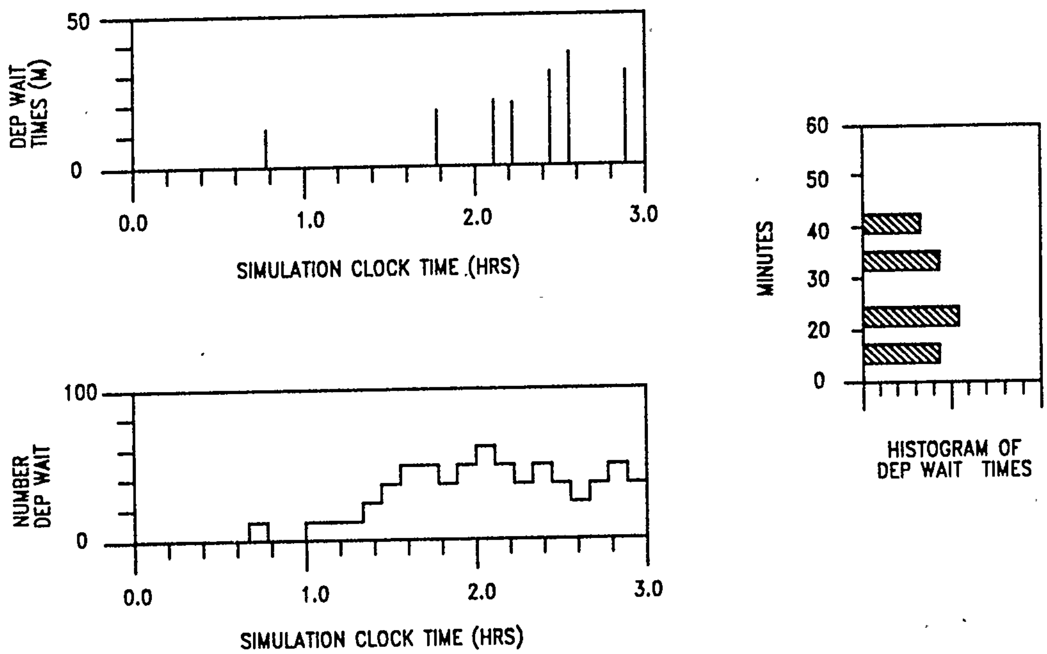
Aircraft Position and Statistics Files are created for post-processing by programs that produce graphical output on Tektronix 4100-series colour displays. At a specified simulation-time interval (usually 30 seconds), the current positions and altitudes of all aircraft are written by TFS to the Aircraft Position File. The TFS Aircraft Position Display program (TFSG) processes this file and the scenario definition files to produce the display shown in Figure 4.6, in which all airspace features - the terminal area boundary, incoming airways, the runway, and fixes - are displayed. (As Figure 4.7 shows, the latitude-longitude grid is optional, and the image may be zoomed.) Aircraft are displayed as coloured triangles enclosing a colour-coded altitude numeral. The triangle colour denotes aircraft type, the numeral colour encodes the 10000s digit of the altitude, and the numeral itself gives the 1000s digit. The TFSG program produces an animated display of aircraft movements by stepping through the Aircraft Position File.

From the Statistics File, the TFS Performance Statistics Display Program (TFSP) produces the statistical graphs shown in Figure 4.8. These graphs may be produced for any approach track, holding fix, or runway that is defined in the simulation airspace; data may be selected for plotting from any time interval in the run.

Figure 4.8a illustrates the graphs produced by TFS for a runway. The top-left graph is a spike plot showing the length of time that each departure waited before receiving departure clearance. A spike is plotted for each departure: the length of the spike shows the waiting time, and its location on the Simulation Time axis shows the time at which the departure started its takeoff. The lower-left graph is a time-series plot of the number of waiting departures, with superimposed, colour-coded event marks that indicate runway occupancy by an arrival or departure. The right-hand graph shows a histogram of depature waiting times. Note that the data plotted in the three graphs is selected from the time period entered by the user in the dialog area of the screen. The dialog area also displays the average number of waiting departures and the average waiting times over the time period.

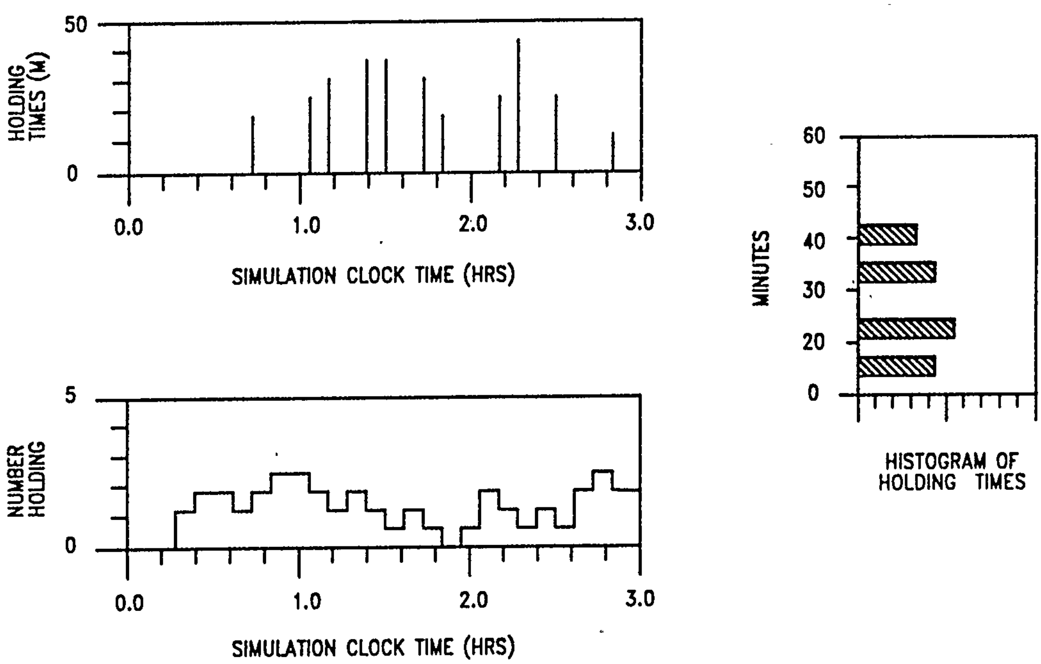
Figure 4.8b illustrates the graphs that are produced for a holding fix. In the top left graph, a spike is plotted at the time an aircraft leaves a holding pattern to begin its approach, and the length of the spike represents the time spent by the aircraft in the hold. The other two graphs show a time-series of the number of aircraft holding at the fix, and a histogram of holding times. The dialog area displays the average number of aircraft holding, and the average holding times over the selected time period.

Figure 4.8c illustrates the graphs that are produced for an approach track. In the top left graph, a spike is plotted at the time that an aircraft lands, and its length represents the time taken by the aircraft to fly the approach. The other two graphs show a time series of the number of aircraft simultaneously flying the approach track, and a histogram of the approach times.



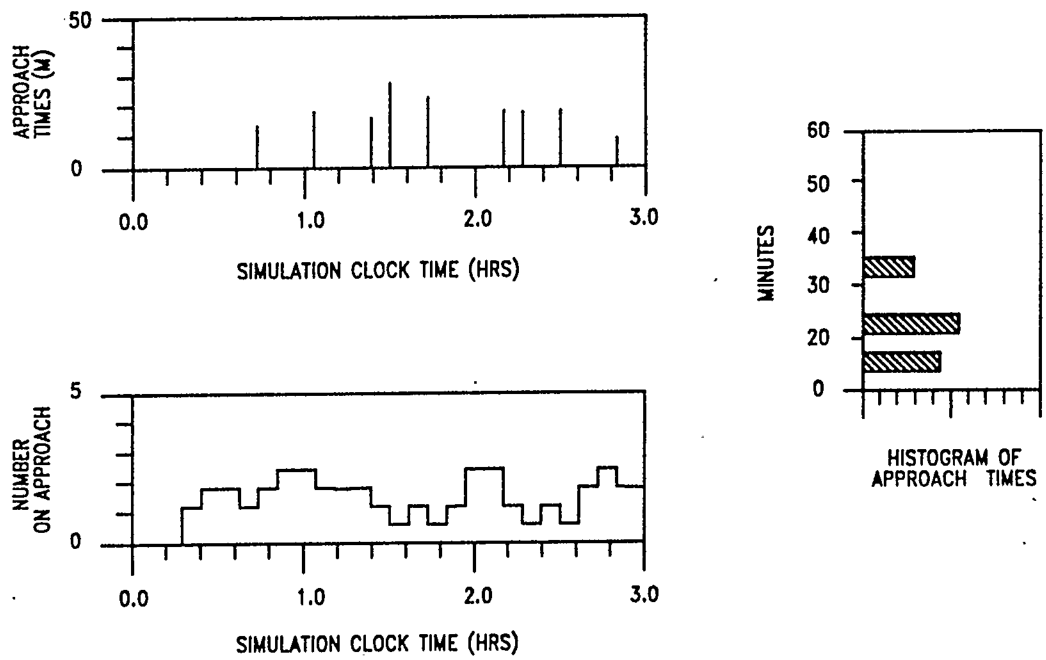
SIMULATION ID:	test01	RUNWAY	CYQT25
REPLICATION #	1	AVERAGE DEP WAIT TIME	20.04
START TIME	0:00:00	AVERAGE NO. WAITING	3.97
END TIME	3:00:00		

Figure 4.8a Runway Statistical Graph



SIMULATION ID:	test01	HOLDING FIX	AA30
REPLICATION #	1	AVERAGE HOLDING TIME	17.08
START TIME	0:00:00	AVERAGE NO. HOLDING	1.71
END TIME	3:00:00		

Figure 4.8b Holding Pattern Statistical Graph



SIMULATION ID:	test01	APPROACH TRACK ID	25RD
REPLICATION #	1	AVERAGE APPROACH TIME	16.44
START TIME	0:00:00	AVERAGE NO. ON APPR	1.44
END TIME	3:00:00		

Figure 4.8c Approach Track Statistical Graph

The dialog area displays the average number of aircraft on the approach and the average approach time.

5. SIMULATION EXPERIMENTS AND RESULTS

This chapter demonstrates the application of the TFS model to an analysis of the effect of arrival "metering" on terminal-area capacity and delay.

Metering is a technique of controlling the rate of entry of arriving aircraft into the terminal area, with the purpose of matching the entry rate to the sustainable landing rate; the sustainable landing rate, or "runway capacity", depends on the separation rules that are in effect and on the approach speeds of the arriving aircraft.

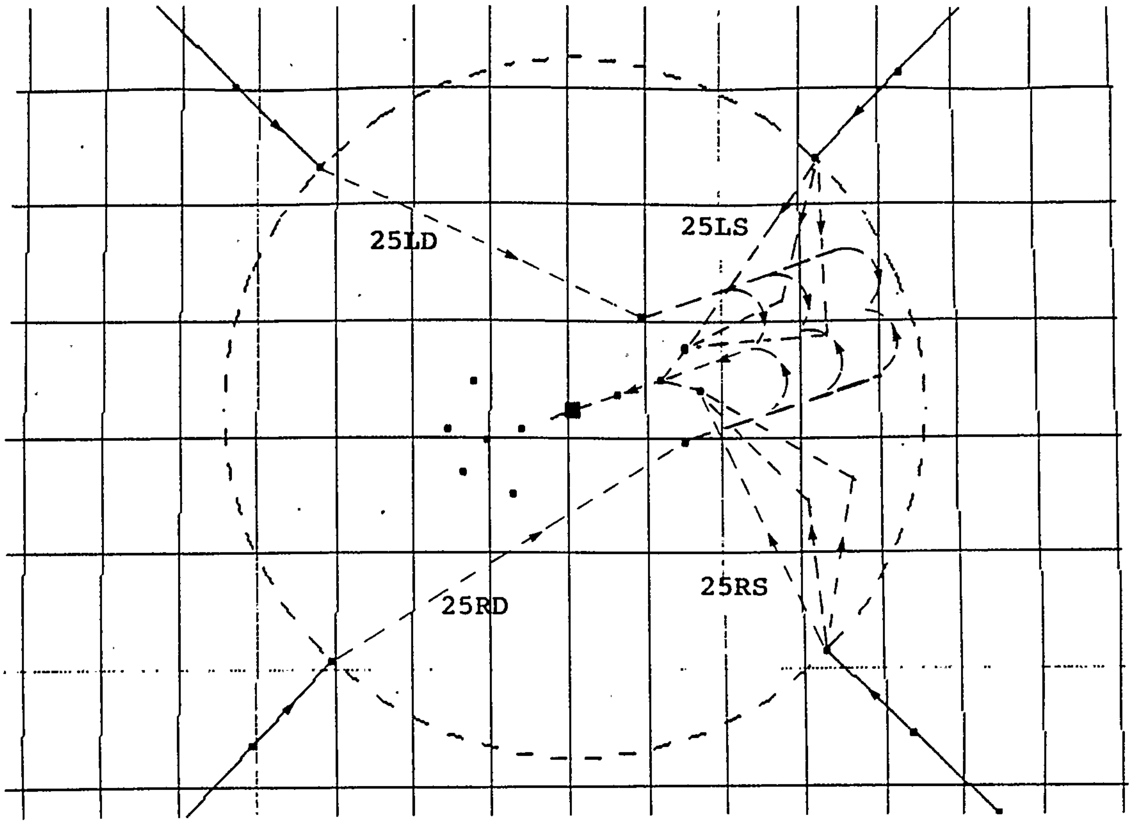
Runway capacity is the basic parameter of the simple metering algorithm modelled here. If runway capacity is specified as n landings per hour, the algorithm divides each hour into n "landing slots", 60/n minutes apart. Each arriving aircraft passes over one of a set of "metering fixes" that are located some distance outside the terminal area; as the aircraft approaches a metering fix, its ETA at the runway is calculated to determine its priority for assignment of a landing slot. In this simple algorithm, the unassigned aircraft with the earliest ETA receives the next landing clearance. Its runway ETA is compared to the next unassigned slot, and if the ETA is equal to or later than that slot, the aircraft is cleared to continue over the metering fix and to enter the terminal area. If the ETA is earlier than the slot, the aircraft is held at the metering fix until an assigned fix-departure time, which is calculated to shift its runway ETA to the next free slot.

Clearly the concept of assigned landing slots is somewhat idealized, since an aircraft's actual landing time will almost certainly deviate from the assigned slot. The time at which the aircraft passes inbound over the metering fix will deviate from the estimated time, and additional error will be introduced into the runway ETA because it is calculated from the average approach time for the aircraft's performance class. The actual approach time will be affected by weather conditions and by vectoring.

The following sections present the results of TFS simulation runs for two sets of flight-plan files: in one set the aircraft inter-arrival times simulate the arrival pattern of metered traffic and the inter-arrival times in the other set represent unmetered arrival traffic. Because computer time is scarce at the R&E Centre, and because it would be costly to develop a scenario that would represent the details of traffic flow and ATC procedures in an actual terminal area, these simulation scenarios are somewhat simplified. The purpose here is to describe a typical application of the model, and to demonstrate some of its features, without undertaking the intensive data gathering and the multiple replications that would be required in an actual simulation exercise.

5.1 The Simulation Scenarios

Figure 5.1 shows a map of the terminal area for the Flow Metering (FM) simulations. The layout of the airspace is identical to the terminal area that was simulated in a recent real-time exercise for an existing Canadian airport (during the planning of the real-time exercise,



SIMULATION ID FM011F

REPLICATION #

TOTAL # AIRCRAFT

AIRCRAFT HOLDING

DEPARTURES WAITING

SIMULATION TIME

NEXT SIM TIME

POSITION INTERVAL

SIM START TIME

SIM END TIME

Figure 5.1 Terminal Area Map for the FM Simulations this TFS airspace was created so that TFS could be used to verify approach track timings).

The terminal area boundary is a circle of radius 30 nautical miles, centred on a VOR located next to the runway. Traffic arrives on four airways, and enters the terminal area at IAFs AA30, BB30, CC30, and DD30; if necessary, traffic is held at these IAFs until an approach track is available. The four approach tracks between the IAFs and the arrivals runway (runway 25) are labelled 25RS, 25LS, 25LD, and 25RD. As indicated in Figure 5.1, the approach tracks all contain vectoring areas; tracks 25RD and 25LD include downwind vectoring areas, and tracks 25RS and 25LS include direct vectoring areas. (The operation of these vectoring areas is demonstrated in a series of figures in section 5.3.)

In the simulation runs reported here, the traffic streams for the simulations were read from flight-plan files created by a program external to TFS. These flight-plan files were created in pairs: one file (the "N" file, for "No flow metering") contains a sequence of flight-plans in which the intervals between successive aircraft flying over a metering fix are drawn randomly from an exponential distribution, and the other file (the "F" file) contains a transformation of the first file in which the intervals have been modified to simulate the effects of flow metering on the "N" traffic stream. The modification consists of adding to the fix-departure time the metering delay calculated by the algorithm described above, plus a random component intended to represent the inaccuracies in arrival estimates and flying times. The hypothetical metering fixes are located on the arrival airways, 50 nautical miles from the terminal centre. TFS creates the aircraft at these points at their calculated fix-departure times.

Each of the "N" files contains the output of four random traffic streams, one for each of the arrival airways. For simplicity, the streams are homogeneous, consisting only of B727 aircraft, and only the average inter-arrival time is varied among the airways. For all of the simulations reported below, the inter-arrival times of the traffic approaching along the 25RD and 25LD tracks is four per hour, and six per hour for the 25RS and 25LS tracks.

Four pairs of files were created, two replications with the metering rate set to 30 per hour (files FM001F, FM001N, FM002F, and FM002N) and two replications with the rate set to 20 per hour (files FM011F, FM011N, FM012F, and FM012N). Aircraft in the metered files enter the airspace within one minute of the assigned fix-leaving time (the deviation is random).

Standard radar separation rules are used in the eight simulations: 3 nautical miles horizontal separation, and 1000' vertical separation.

5.2 Simulation Results

The results of the simulations are summarized in Tables 5.1a to 5.1d, and Figures 5.2a to 5.2d, which show the effect of flow metering on delay for each of the four pairs of simulations.

The tables and figures present descriptive statistics for six variables calculated for each aircraft in the flight-plan files:

- EntryDelay is the difference between the time that the aircraft passed the metering fix inbound in the "N" simulation and the time that it left the fix in the "F" run; it measures the delay assigned to the aircraft by the flow-metering algorithm.

- LandDelay is the difference in the aircraft's landing times in the two runs; if metering is reducing the amount of holding time and vectoring delay in the terminal area, LandDelay should be less than EntryDelay.

- F_HoldTime and N_HoldTime measure the length of time that the aircraft was flying a holding pattern in the "F" and "N" runs, respectively.

- F_ApprTime and N_ApprTime measure the interval between the time that the aircraft passed its IAF inbound and the time that it landed; vectoring delays will lengthen these intervals.

Table 5.1 presents the sums, averages and maximum values of these six variables for each of the four pairs of simulations. Several conclusions may be drawn from the table:

- In the two simulation pairs (FM001F/N and FM002F/N) for which the metering-rate parameter was set to 30 landings per hour, flow metering has little benefit. The average time that aircraft spend in holding patterns at the IAFs is, in fact, slightly longer in the FM002F simulation than in FM002N. Added to the average entry delay of more than two minutes in FM001F and FM002F, flow metering does more harm than good when the rate is set this high. (Actually, later tests with "saturation" traffic rates estab-

ApproachTrack	25LD	25LS	25RD	25RS	ALL
SUM EntryDelay	52.20	29.66	29.53	39.26	150.65
SUM LandDelay	7.90	11.82	23.57	39.64	82.93
SUM F_HoldTime	34.85	38.09	31.88	42.82	147.64
SUM N_HoldTime	74.55	58.11	48.79	46.42	227.87
SUM F_ApprTime	188.72	202.12	193.73	163.79	748.36
SUM N_ApprTime	193.44	202.47	182.69	158.37	736.97
AVG EntryDelay	4.02	1.85	2.27	2.45	2.60
AVG LandDelay	0.61	0.74	1.81	2.48	1.43
AVG F_HoldTime	2.68	2.38	2.45	2.68	2.55
AVG N_HoldTime	5.73	3.63	3.75	2.90	3.93
AVG F_ApprTime	14.52	12.63	14.90	10.24	12.90
AVG N_ApprTime	14.88	12.65	14.05	9.90	12.71
MAX EntryDelay	9.83	10.68	8.25	8.94	10.68
MAX LandDelay	3.47	3.31	4.98	7.59	7.59
MAX F_HoldTime	7.08	7.33	9.65	7.38	9.65
MAX N_HoldTime	15.33	13.82	13.68	13.76	15.33
MAX F_ApprTime	17.58	13.82	18.08	11.89	18.08
MAX N_ApprTime	17.67	13.90	17.51	11.92	17.67
Number of Aircraft	13	16	13	16	58

Table 5.1a: FM001 Summary of Delay, Holding, and Approach Times

ApproachTrack	25LD	25LS	25RD	25RS	ALL
SUM EntryDelay	29.59	21.61	31.36	44.62	127.18
SUM LandDelay	38.53	23.44	62.59	29.48	154.04
SUM F_HoldTime	47.25	46.52	51.59	57.08	202.44
SUM N_HoldTime	39.81	44.18	36.66	63.41	184.06
SUM F_ApprTime	160.94	158.75	249.05	202.95	771.69
SUM N_ApprTime	159.90	160.39	232.80	214.25	767.34
AVG EntryDelay	2.69	1.66	1.96	2.12	2.08
AVG LandDelay	3.50	1.80	3.91	1.40	2.53
AVG F_HoldTime	4.30	3.58	3.22	2.72	3.32
AVG N_HoldTime	3.62	3.40	2.29	3.02	3.02
AVG F_ApprTime	14.63	12.21	15.57	9.66	12.65
AVG N_ApprTime	14.54	12.34	14.55	10.20	12.58
MAX EntryDelay	7.32	5.30	7.00	8.08	8.08
MAX LandDelay	8.45	12.21	11.67	8.59	12.21
MAX F_HoldTime	14.70	11.75	11.60	10.08	14.70
MAX N_HoldTime	13.48	9.88	11.48	9.88	13.48
MAX F_ApprTime	18.87	13.62	18.96	11.72	18.96
MAX N_ApprTime	17.58	13.50	20.01	11.17	20.01
Number of Aircraft	11	13	16	21	61

Table 5.1b: FM002 Summary of Delay, Holding, and Approach Times

ApproachTrack	25LD	25LS	25RD	25RS	ALL
SUM EntryDelay	123.77	111.36	201.28	265.08	701.49
SUM LandDelay	97.51	78.24	175.35	219.88	570.98
SUM F_HoldTime	5.25	6.66	2.27	0.00	14.18
SUM N_HoldTime	37.61	38.15	26.92	44.83	147.51
SUM F_ApprTime	149.80	145.28	185.59	165.37	646.04
SUM N_ApprTime	145.42	149.04	189.20	172.06	655.72
AVG EntryDelay	12.38	9.28	15.48	15.59	13.49
AVG LandDelay	9.75	6.52	13.49	12.93	10.98
AVG F_HoldTime	0.53	0.56	0.17	0.00	0.27
AVG N_HoldTime	3.76	3.18	2.07	2.64	2.84
AVG F_ApprTime	14.98	12.11	14.28	9.73	12.42
AVG N_ApprTime	14.54	12.42	14.55	10.12	12.61
MAX EntryDelay	23.22	22.47	22.10	24.08	24.08
MAX LandDelay	21.00	21.23	22.21	20.34	22.21
MAX F_HoldTime	3.08	2.25	2.27	0.00	3.08
MAX N_HoldTime	13.48	9.88	11.48	9.88	13.48
MAX F_ApprTime	17.08	13.59	15.83	11.77	17.08
MAX N_ApprTime	17.58	13.50	20.01	11.17	20.01
Number of Aircraft	10	12	13	17	52

Table 5.1c: FM011 Summary of Delay, Holding, and Approach Times

ApproachTrack	25LD	25LS	25RD	25RS	ALL
SUM EntryDelay	287.22	162.50	241.80	152.21	843.73
SUM LandDelay	237.06	138.38	188.46	122.05	685.95
SUM F_HoldTime	21.67	13.76	0.00	10.55	45.98
SUM N_HoldTime	75.45	42.97	40.70	48.63	207.75
SUM F_ApprTime	227.11	172.24	195.61	73.56	668.52
SUM N_ApprTime	224.90	169.09	209.56	67.57	671.12
AVG EntryDelay	19.15	11.61	18.60	21.74	17.22
AVG LandDelay	15.80	9.88	14.50	17.44	14.00
AVG F_HoldTime	1.44	0.98	0.00	1.51	0.94
AVG N_HoldTime	5.03	3.07	3.13	6.95	4.24
AVG F_ApprTime	15.14	12.30	15.05	10.51	13.64
AVG N_ApprTime	14.99	12.08	16.12	9.65	13.70
MAX EntryDelay	32.55	32.95	32.40	32.25	32.95
MAX LandDelay	26.76	26.45	26.91	29.16	29.16
MAX F_HoldTime	7.78	3.90	0.00	6.13	7.78
MAX N_HoldTime	15.41	11.58	15.33	13.67	15.41
MAX F_ApprTime	19.42	13.60	18.17	11.92	19.42
MAX N_ApprTime	19.92	13.62	19.92	11.08	19.92
Number of Aircraft	15	14	13	7	49

Table 5.1d: FM012 Summary of Delay, Holding, and Approach Times

lished that the capacity of the TFS ATC function for this airspace is about 20 landings per hour, so these results are not surprising.)

. In the two simulation pairs (FM011F/N and FM012F/N), however, the metering rate was set more realistically at 20 landings per hour, and the benefits of flow metering may be seen in the substantial reduction in the time that aircraft spend holding at IAFs. Over two hours of holding time is saved in each of the two metered simulations, with an average saving of 2 1/2 to 3 minutes per aircraft, and a reduction of 8 to 10 minutes in maximum holding time. The cost is an average entry delay of 13 to 17 minutes, which may seem to substitute a longer period of holding at the metering fix for holding at the IAF. However, if ATC gives a pilot advanced warning of his metering delay (as it usually does), he may absorb the delay by reducing speed en route or by waiting on the ground before departing on a flight to the flow-metered terminal area. Both of these alternatives are much less expensive than holding at the IAF. In fact, even a hold at the metering fix is less expensive than a hold at the IAF, since aircraft at the metering fix hold at a higher altitude where fuel consumption is lower.

. There is no consistent difference in approach time between the metered and unmetered runs. Flow metering does not permit the ATC function to consistently assign shorter approach paths to arriving aircraft because the traffic rate in these simulations is close to the TFS capacity for this airspace, and ATC constantly uses vectoring delays to merge the

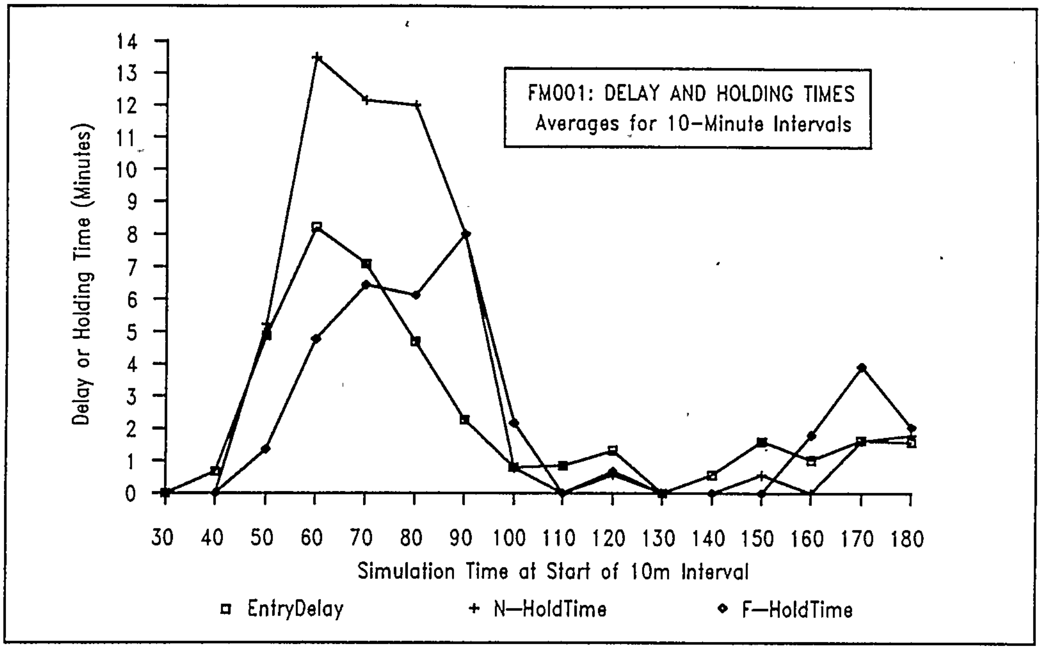


Figure 5.2a

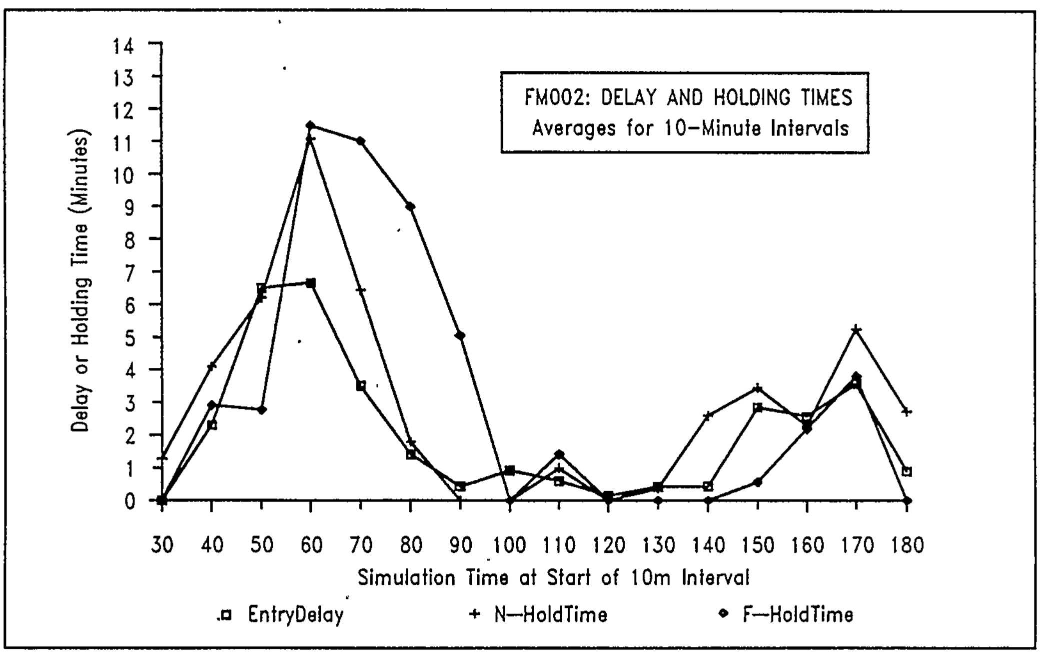


Figure 5.2b

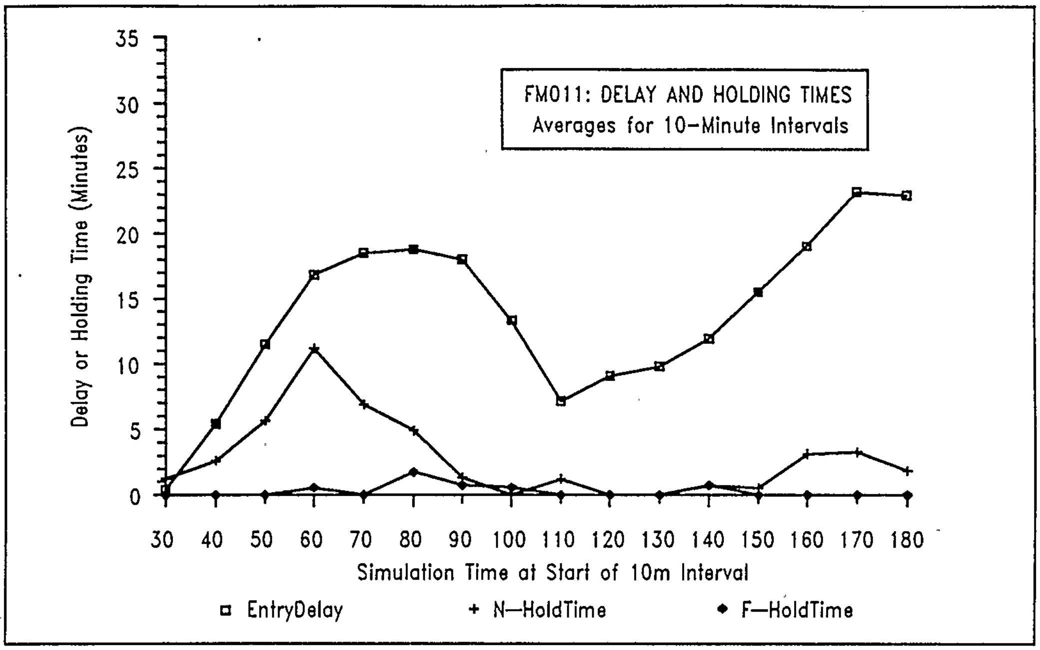


Figure 5.2c

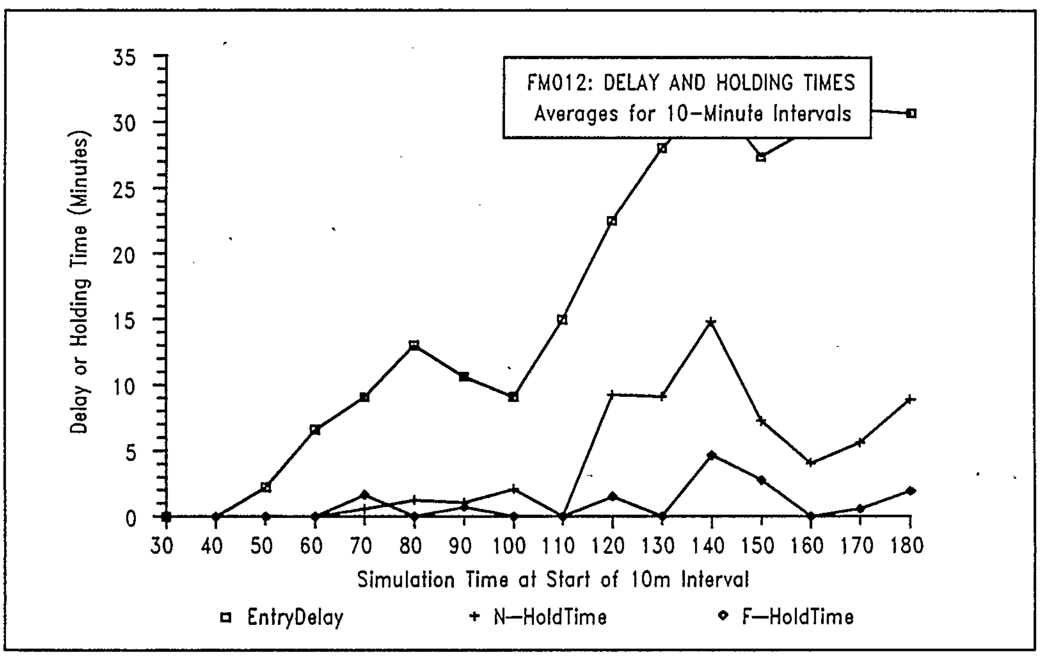


Figure 5.2d

traffic into an approach sequence. The main benefit of flow metering, therefore, is the reduction of holding times.

Figure 5.2 plots the average values of EntryDelay, N_HoldTime, and F_HoldTime for the aircraft that landed in each 10-minute interval of simulation time. (Landing time was arbitrarily taken from the "F" simulation.) The plots are consistent with the results in Table 5.1, and show that holding times in the flow-metered runs are substantially less in the two simulation pairs with realistic metering rates. In these simulations, the metering delays increase steadily over the three-hour simulation period, fluctuating with peaks and valleys in the traffic rate. The effect of metering can be seen clearly in the difference between the holding time for the metered and unmetered runs: the IAF holding time in the "N" runs rises quickly when arrival bursts occur, but in the "F" runs holding time is controlled by metering. (For comparison, Figure 5.3 shows the number of aircraft crossing the four metering fixes in each 10-minute interval in the unmetered flight-plan files. Note the initial bursts of traffic in three of the four simulation pairs, which account for the long delays at the 1-hour mark in these runs.)

5.3 Illustration of Aircraft Vectoring

Figures 5.4a to 5.4c present a sequence of aircraft position displays from simulation FM011F that illustrate the operation of the vectoring areas. In the initial display at 1:28:30, five aircraft are in the terminal area, and one is about to enter from the northeast airway. One aircraft has just turned onto final approach

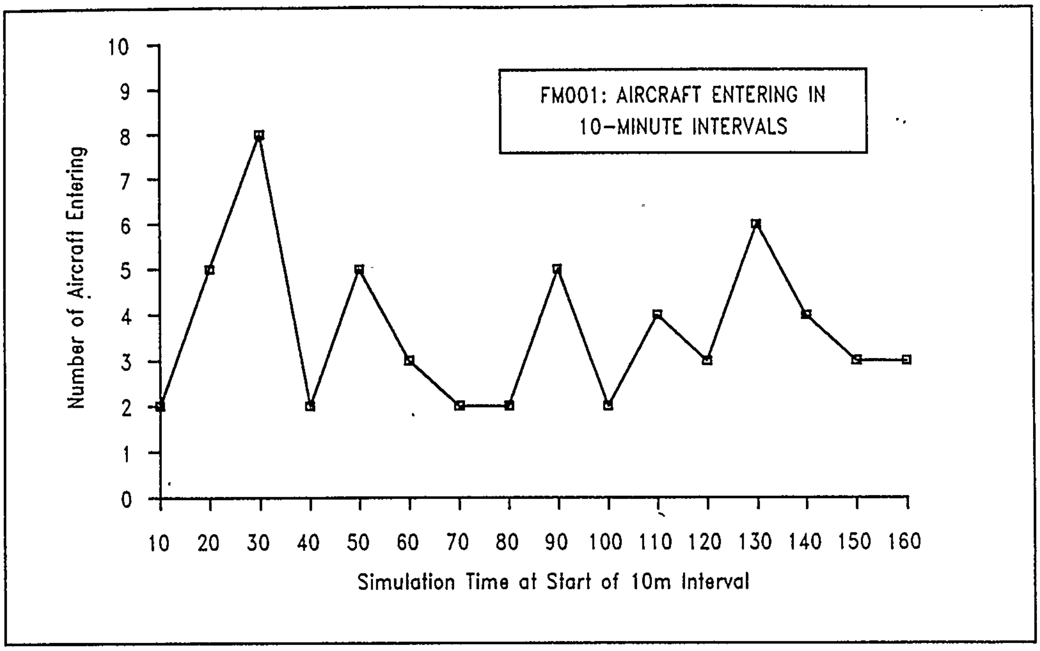


Figure 5.3a

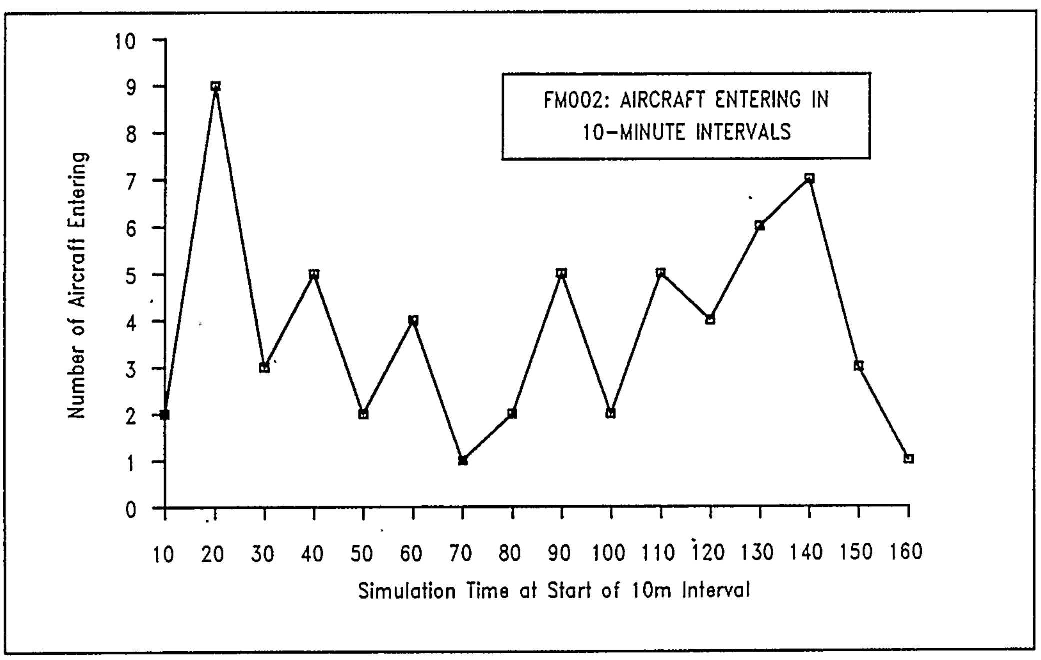


Figure 5.3b

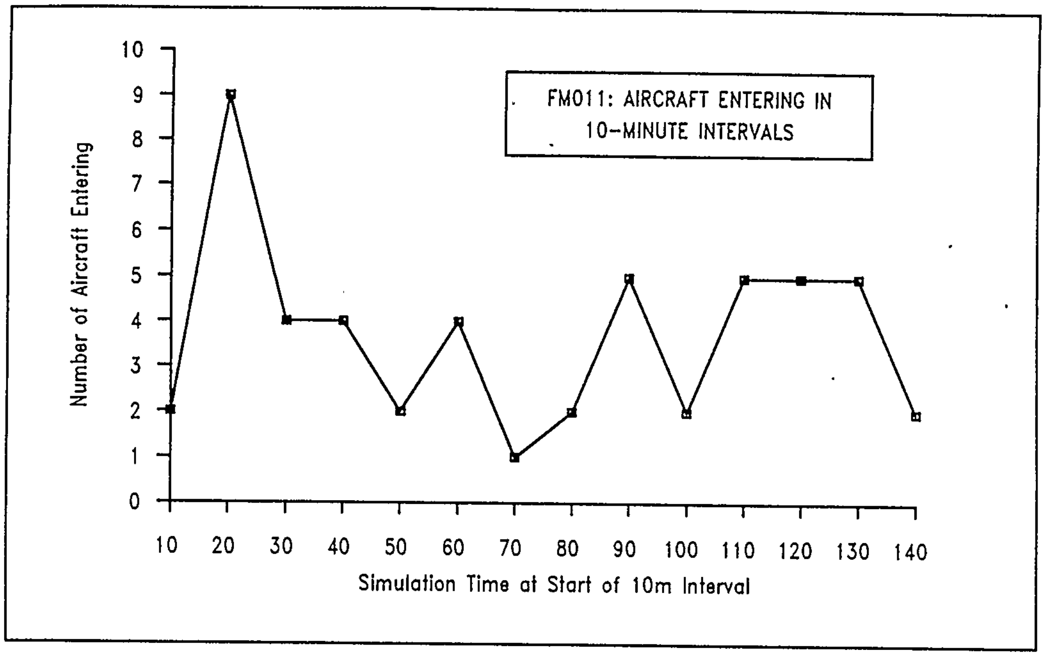


Figure 5.3c

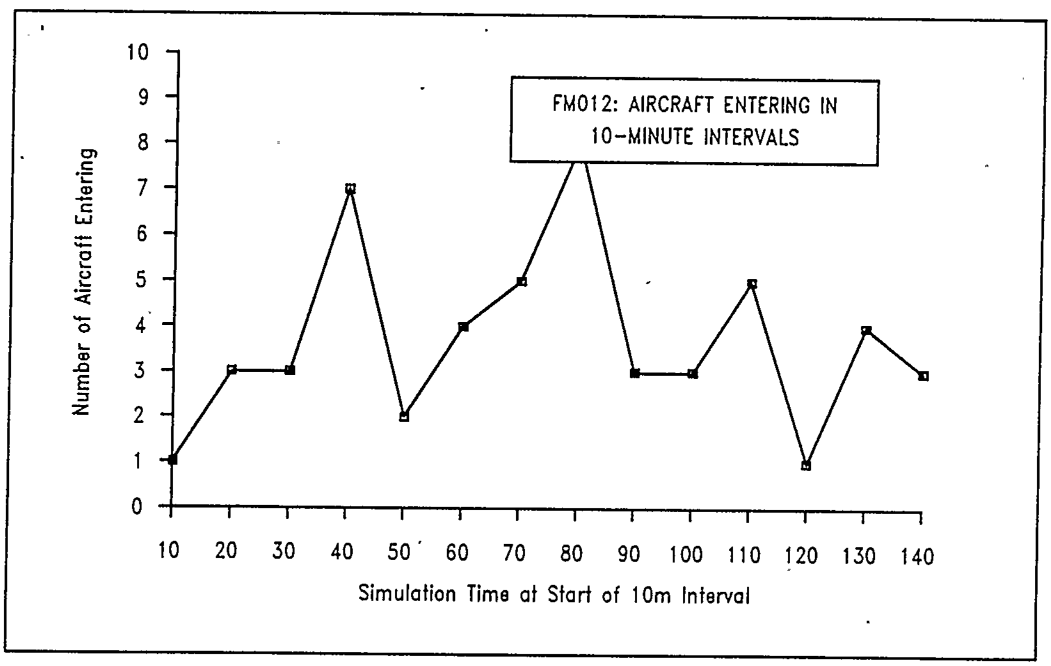


Figure 5.3d

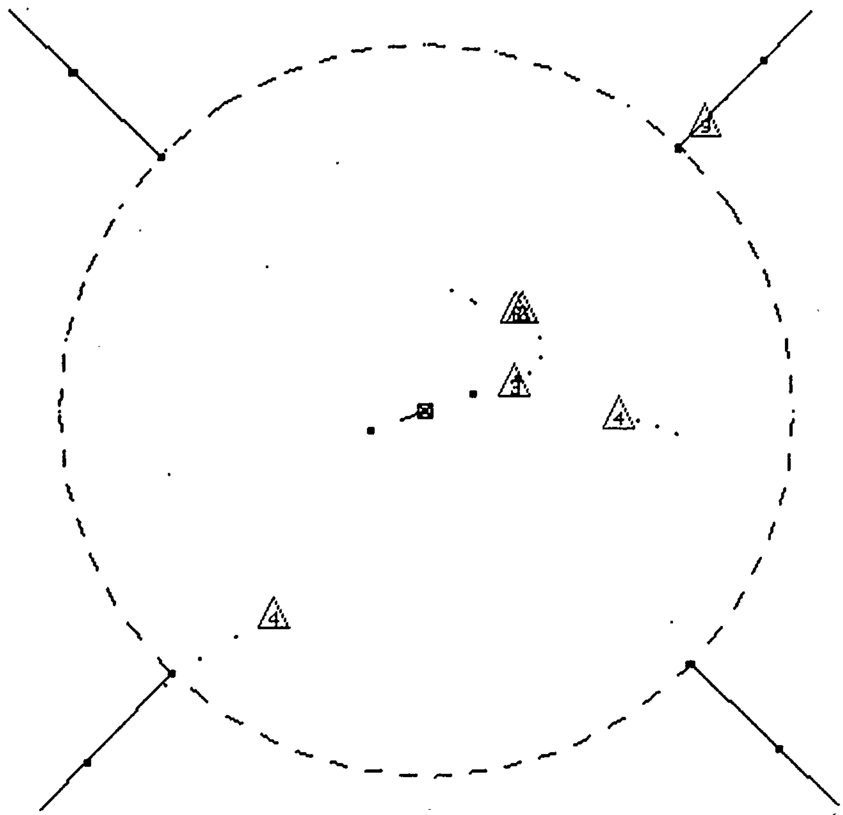
at 3000' altitude, another at 4000' is flying to intercept the localizer after being vectored northward from the southeast fix by ATC, and two aircraft that appear superimposed in the display are at 6000' and 8000' on a downwind vector.

In the display at 1:30:30, the purpose of the vectoring becomes clear, as the two superimposed aircraft are now separated horizontally: the lower aircraft is now at 5000' and has been cleared by ATC to turn and intercept the localizer, while the higher aircraft is at 6000' and is continuing to fly downwind. The aircraft entering from the northeast fix is being diverted on a southward vector so that it will be separated from the leading traffic when it turns into the approach stream.

The display at 1:32:00 shows the net result of the vectoring: five aircraft are aligned on final approach, with adequate horizontal separation. The aircraft flying downwind at 9000' will eventually be turned to intercept the localizer behind the other traffic.

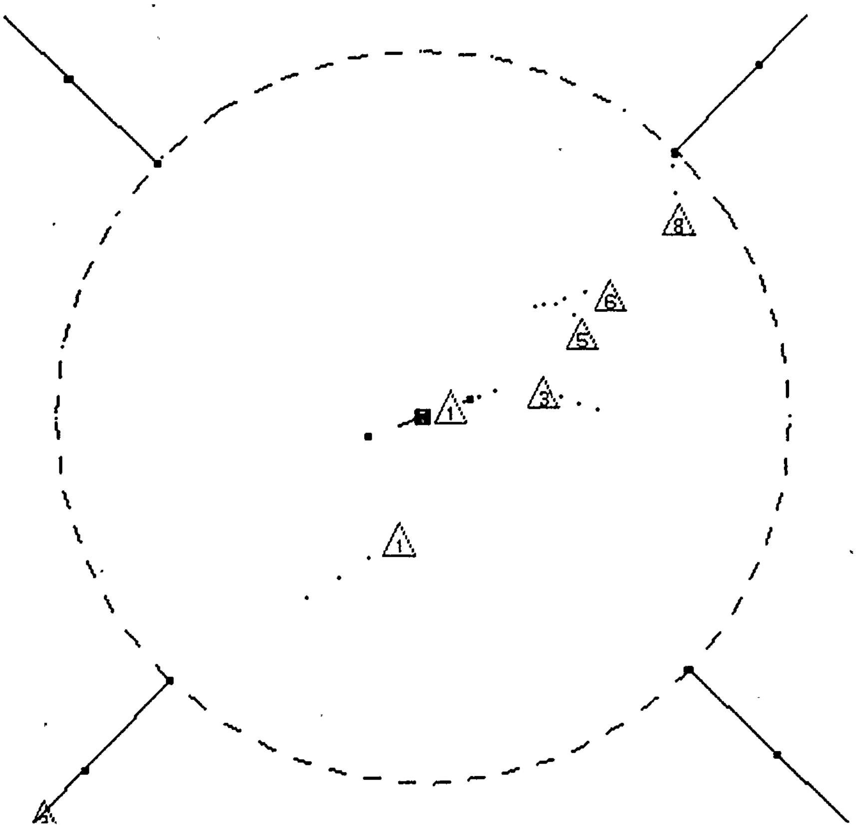
This set of figures illustrates a situation in which ATC uses vectoring to lengthen the approach tracks of selected aircraft in order to create a sequence of arrivals in which the aircraft are properly separated, but in which there are no large gaps that would waste runway capacity.

This sequence is taken from one of the "F" simulations, in which the traffic flow was metered. Unmetered traffic is much more likely to arrive in bursts which may exceed ATC's ability to maintain separation by vectoring; the effects of traffic bursts were seen in the holding delays tabulated in the results for the "N" simulations in the last section.



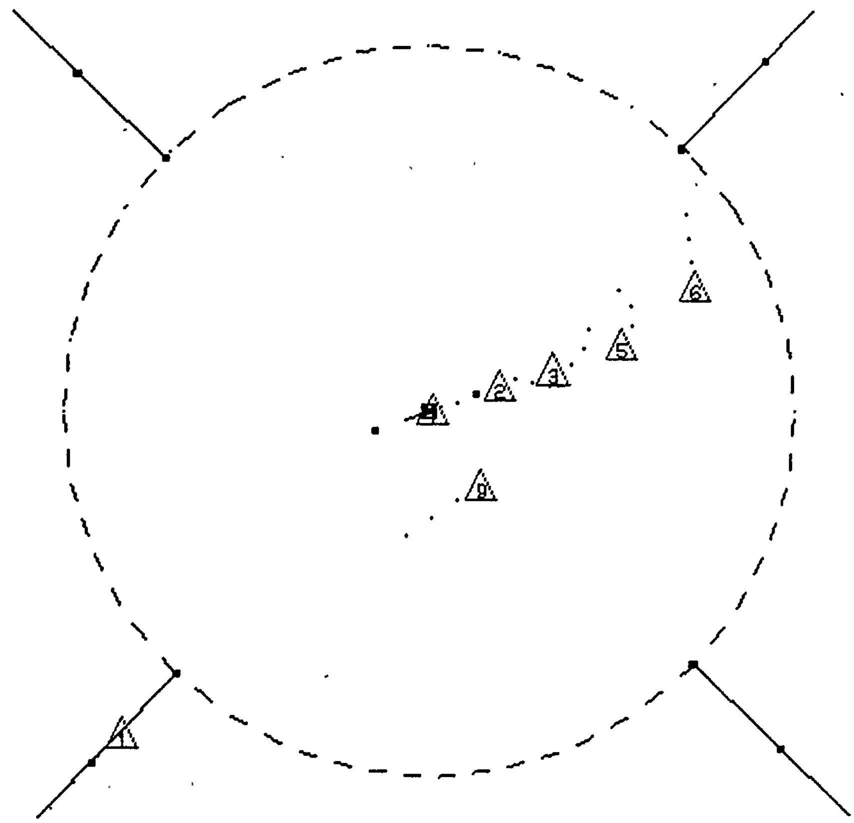
SIMULATION ID	FM011F	SIMULATION TIME	1:28:30
REPLICATION #	1	NEXT SIM TIME	1:29:00
TOTAL # AIRCRAFT	6	POSITION INTERVAL	0:30
# AIRCRAFT HOLDING	0	SIM START TIME	0:00:00
# DEPARTURES WAITING	0	SIM END TIME	3:44:30

Figure 5.4a



SIMULATION ID	FM011F	SIMULATION TIME	1:30:30
REPLICATION #	1	NEXT SIM TIME	1:31:00
TOTAL # AIRCRAFT	7	POSITION INTERVAL	0:30
# AIRCRAFT HOLDING	0	SIM START TIME	0:00:00
# DEPARTURES WAITING	0	SIM END TIME	3:44:30

Figure 5.4b



SIMULATION ID	FM011F	SIMULATION TIME	1:32:00
REPLICATION #	1	NEXT SIM TIME	1:32:30
TOTAL # AIRCRAFT	7	POSITION INTERVAL	0:30
# AIRCRAFT HOLDING	0	SIM START TIME	0:00:00
# DEPARTURES WAITING	0	SIM END TIME	3:44:30

Figure 5.4c

6.0 DISCUSSION AND CONCLUSIONS

This thesis has described the development of a prototype fast-time air-traffic simulation model for the Research and Experimentation Centre of Transport Canada's Air Traffic Services; the model is referred to as the TFS ("Terminal-Area Fast-Time Simulation") model. The TFS software includes data-entry programs for the definition of airspace structure, the TFS simulation program itself, and graphics display programs. The current version of this software is capable of modelling air-traffic movement arriving and departing from a single-runway airport in a terminal-area airspace that is under radar or procedural control.

The TFS model has already been used in actual planning projects, including a cost/benefit study of radar installations, in which the model's ability to simulate both radar and procedural control was used to estimate the effect on delay and capacity of installing a surveillance radar in a terminal area. The applications of the model in these projects, as well as the extensive testing and experimentation that was conducted during the development of TFS, has led to number of conclusions regarding the design and performance of the model. These conclusions are presented in the following section. Also, because of known requirements for fast-time simulation studies in the near future, a number of extensions to the model will be needed to extend its applicability to large, multi-runway terminal areas. These extensions are discussed in Section 6.2, as are long-range plans for developing the model.

6.1 Evaluation of Current Model

Air-traffic controllers at the R&E Centre have studied the graphic display of TFS traffic flows and made subjective evaluations of the degree of realism of the traffic patterns. Allowing for the acknowledged limitations of the initial version of TFS, they have judged the simulated traffic flow to be realistic enough to be useful in airspace planning studies. The controllers have criticized specific aspects of the ATC sub-model of TFS, and their criticisms will be addressed in future versions of the model.

Traffic capacity provides a more objective measure of the TFS model's performance. In the terminal-area simulations described in Chapter 5, the simulated radar-controlled terminal area could accept about twenty arrivals per hour, even though it was operating with a relatively limited set of vectoring options. It seems likely that this initial version of the model can be improved so that its performance is reasonably close to the observed capacity of actual terminal areas (25 to 30 arrivals per hour for single-runway operations at a large international airport).

Several conclusions were also drawn regarding the design approach and software implementation of TFS . First of all, the decision to use SIMULA as the implementation language was completely justified by the programming productivity that was achieved in the development of TFS. The TFS program includes many examples of complicated aircraft manoeuvres and ATC procedures which are concisely represented in SIMULA as sequences of timed events and state events. (While the DISCO subclasses also contributed greatly to productivity in the combined discrete-continuous simulation of aircraft motion, the generality of the DISCO facilities is reflected in processing time. Future versions of TFS will include more specialized discrete-continuous procedures, tailored to the specific TFS requirement.)

Secondly, experience during the project also justified the use of discrete-continuous simulation methods to model the motion of individual aircraft. This approach was chosen in part because it would support an animated situation display of the airspace, and this display, a unique feature of TFS, has amply demonstrated the value of animated graphical presentation of simulation results. As well as providing a useful debugging tool (the display programs were operational in the very early stages of TFS development), the display of aircraft movements has allowed air-traffic controllers to evaluate the performance of the model and to provide detailed guidance and feedback to the TFS project team throughout the development process. Many of the usual problems of communication between users and developers have therefore been avoided.

6.2 Future Directions

In response to user suggestions, and to known requirements for future fast-time simulation studies, the following directions have been identified for the development of TFS:

- Addition of variability in the components of the "control loop": pilot errors and reaction times, navigation system errors, communication delays, and ATC reaction times.

- Development of a weather sub-model, and modification of the aircraft motion and ATC processes to include weather factors.

- Development of terminal-area ATC functions to simulate the adjustment of arrival separation in response to the length of the departure queue; the simultaneous use of several arrival and departure runways; the re-assignment of active runways; the operations of a control area containing several airports; issuing of approach clearances to aircraft executing a missed-approach procedure; and the use of speed control to maintain separation on approach tracks.

- Development of program components for the modelling of sectorized airspaces: components that will model air-traffic control operations and aircraft movements in enroute sectors, including the communications between enroute and terminal area controllers; an interface component in the terminal-area model that will accept arrivals from, and hand departures off to, the enroute sector; and an integrated flow-control model that will simulate the operation of metering fixes.

With these extensions, the TFS model will be applicable to virtually any Canadian airspace, and will provide a cost-effective fast- time simulation facility to complement Transport Canada's existing real-time IFR simulation facilities.

REFERENCES

1. Canada. Department of Transport. Instrument Procedures Manual (TP2076). September 1981.

2. Canada. Department of Transport, A.I.P Canada: Aeronautical Information Publication (TP2300). January 1984.

3. Canada. Department of Transport, Air Traffic Control Manual of Operations (TP703). August 1983.

4. Barrer, J.N. and Weiss, W.E. FAA Requirements for Models to Support Capacity, Delay, Noise, and Energy Programs. FAA Report DL5-85-1, March 1985.

5. Canada. Department of Transport. Draft Specification for Flight Data Processing System (FDMP 0001). June 1986.

6. Billman, B., Spracklin, D., and Thomas, J. "Application of Fast-Time Discrete Simulation Techniques in Evaluating Aircraft Collision Algorithms". Proceedings of the Fourteenth Annual Simulation Symposium, March 1981.

7. Schnuerer, H. Monte Carlo Simulations of VOR/DME Holding Procedures. Deutsche Forschungs- und Versuchsanstalt fuer Luft- und Raumfahrt (DLR) Report FB-77-08 (NTIS N77-30105), March 1977.

8. Fricke, M. and Hormann, A. Simulation Analysis and Derivation of Accuracy Requirements for Future Terminal-Area Navigation. Part I: Single-Aircraft Analysis. Bundesministerium fur Forschung und Technologie (BMFT) Report FB-W 83-036 (NTIS N84-2156), December 1983.

9. Fricke, M. and Hormann, A. Simulation Analysis and Derivation of Accuracy Requirements for Future Terminal-Area Navigation. Part II: Multi-Aircraft Analysis. BMFT Report FB-W 83-035 (NTIS N84-21535), December 1983.

10. Odoni, A.R. and Simpson, R.W. Review and Evaluation of National Airspace System Models. FAA Report EM-79-12, October 1979.

11. Bobick,J.C. and Couluris,G.J. Airport and Airspace Delay Model Description. FAA Report AVP-79-11, October 1979.

12. The Aerospace Corporation. Terminal Area Air Traffic Control Simulation Final Report. NASA Report CR-152017, June 1977.

13. Franta, W.R. The Process View of Simulation. New York: North-Holland, 1978.

14. Helsgaun, Keld. "DISCO - A SIMULA-based Language for Continuous Combined and Discrete Simulation". Simulation. July 1980, 1-12.

15. Birtwistle, G.M., Dahl, O-J., Myhrhaug, B., and Nygaard, K. SIMULA Begin. Lund: Studentlitteratur, 1979.

16. Dahl, O-J., Myhrhaug, B., and Nygaard, K. SIMULA 67 Common Base Language. Oslo: Norwegian Computing Centre, 1971.

17. Inkster, James, Lomow, Greg, and Unger, Brian. "Combined Discrete and Continuous Simulation in Ada". Simulation in Strongly Typed Languages: Ada, Simula, Pascal. Proceedings of the Conference on Simulation in Strongly Typed Languages. La Jolla: Simulation Councils Inc, 1984.

18. Pritsker, A.A.B., and Pegden, C.D. Introduction to Simulation and SLAM. New York: John Wiley, 1979.

19. Mitchell, E.L. and Gauthier, J.S. "Advanced Continuous Simulation Language (ACSL)". Simulation. March 1976, 72-78.

20. Birtwistle, Graham. DEMOS - A System for Discrete Event Modelling on Simula. London: MacMillan, 1979.

21. Buhr, R.J.A. System Design with Ada. Englewood Cliffs, NJ: Prentice-Hall Inc., 1984.