A NETWORK MODEL OF THE U. S. AIR TRANSPORTATION SYSTEM

Aurel N. de Hollan
Arthur S. Priver
Andres Zellweger
Transportation Systems Center
Department of Transportation
Cambridge, Massachusetts 02142

Abstract

A basic conceptual simulation of the entire Air Transportation System was developed to serve as an analytical tool for studying the interactions among the System elements. The simulation is designed to function in an interactive computer graphics environment which permits rapid alteration of rules and parameters, as well as continuous real-time graphical monitoring of system operations. The simulation described here is the first member in an evolving hierarchy of increasingly complex models, progressing in the direction of closer approximation to the real-world Air Transportation System.

Introduction

All major modes of transportation are basically similar. They involve the movement of vehicles containing people or goods through more or less fixed networks, with each vehicle being piloted by a human (or conceivably by a computer). The network flow is controlled passively by rules and regulations and actively by some centralized or distributed control agency. The function of active control is twofold: maintaining safety, primarily by assisting the pilots in avoiding the nasty two vehicle interaction known as collision, and expediting traffic by management of multi-vehicle situations.

An example is the road network of a large city. The vehicles are automobiles, buses and trucks. They are piloted by drivers and carry passengers or goods. Passive control is exercised by traffic laws and road signs, decentralized active control by traffic lights and human traffic directors. In air transportation, aircraft, piloted through a network of terminals and airway links, carry the passengers and goods. Again, we have passive control by flight regulations and active control by the semicentralized air traffic control (ATC) system.

One interesting fact about transportation is the shift in emphasis from vehicle to pilot to control. The first factor limiting use of automobile transportation was the existence of a reliable and well-functioning vehicle. As reliable and faster cars got on the road, competent drivers became necessary to use them efficiently and safely. When automobile usage became almost universal, coordination and control was required and human traffic directors and their mechanized counterparts had to be introduced.

As the system became overloaded, there was a shift within the control area from tactical control like traffic lights to strategic control mechanisms such as radio traffic advisory reports, switching roads, bridges and tunnels one way according to demand, or even restrictions on the number of vehicles entering an urban area. The air transportation story is analogous to the automobile story, with the last shift from tactical to strategic control evidenced by flow control (which today means not letting aircraft take off if their destination airport is congested).

Two reasons often given to justify computerized simulation of any system are well applicable to transportation systems, especially to those in their mature, control-determined stages. First, the systems are so complex that analytical methods cannot be used to predict or evaluate system level behavior. Analytical methods might give insight into single vehicle behavior, but such behavior is often counter-intuitively related to system behavior. Simulation is the tool that gives reasonably accurate predictions of the behavior of a system with hundreds or thousands of vehicles, sometimes even if the functions used by the simulation to describe single vehicle behavior are poor estimates. The second reason for transportation simulation is that the best way of finding out things about a system, i.e. by experimentation, is far too expensive and politically unfeasible. For instance, there is a gadget called the proximity warning indicator (PWI) that is generally thought to make aircraft safer and, if installed on all aircraft, could mean elimination of some of the chores of ground-based air traffic controllers. But this can't be proven by installing just a few PWIs and watching what happens to the aircraft and the controllers; all the aircraft must have PWIs for conclusive proof. A PWI based control system would be resisted by the aircraft owners who don't want to pay for the gadgets, and, possibly, by the air traffic controllers who are afraid of losing their jobs. So the government must be very sure of the effects of PWI and any other proposed "improvement" on the air transportation system (ATS) before forcing it on the system. Simulation is a good step in making sure.

The simulation reported here is the first in an evolving hierarchy of increasingly complex ones with the eventual goal being a simulation of most of the U.S. ATS. The simulations are based on the unified concept of transportation described in the opening paragraphs. They are or will be capable of analyzing system performance in response to changes in operating rules, control procedures, network configuration, and aircraft type distribution. A typical problem posed to the simulation might be: How much safer than the present ATS is one in which there are 68% more flights per day, sectors (the geographical units into which ATC is distributed, essentially the areas controlled by one controller team) are twice as large, and ARTS[1] (computerized assistance to human air traffic controllers) is operational?

At present there are no ATS simulation capable of answering such questions.[2,3] Most ATS simulations either address themselves only to one subsystem of ATS and have rather skimpy representations of the rest or they are enormous computer programs that run on giant computers. These large programs are so expensive to run that they preclude the necessary experimentation and tinkering required to make any simulation into a useful tool.

Our present or level 1 model, while working extremely well, is by no stretch of the imagination adequate. The conceptual model we have produced simulates a hypothetical ATS somewhat more limited than that of the continental U.S. but it omits no important details and we are convinced that it can be expanded to a full system simulation.

Rather than working our way through a series of detailed subsystem models and pasting them together as they are completed, we chose to work from the top down and started by defining what any transportation system model should have and then proceeded to build such a model for ATS. This makes some of our approximations to certain subsystem functions look rather strange, but we had a very orderly development phase and avoided having to kluge together incompatible subsystem models. The vehicle we have chosen for our model is a fast time interactive simulation on a small computer driven assemblage of displays and other peripheral devices.

The Simulation

In this section we will discuss briefly what the simulation model does and in somewhat more detail how it does it.

The model permits the user to define an arbitrary network configuration, consisting of a set of terminals and navigational aids (VORTAC radio beacons) with specified airway geometry connecting them. Distributed ATC may be defined in the form of terminal (within about 40 miles of takeoff and landing) and enroute sectors.

A schedule may be imposed upon such a network, consisting of predefined flights of specified aircraft types, originating at specified nodes, following specified flight paths with altitudes assigned by the controllers, and landing at specified destinations. Only instrumented (IFR) flights are considered. Multiple flights may be scheduled for departure at the same time at one terminal. Such flights are then sequenced, taking into consideration the number of incoming flights and rules governing aircraft separation in the terminal area. After takeoff, aircraft follow Standard Instrument Departure (SID) routes until their transition to the enroute airway structure. Detailed aircraft dynamics are not modeled, but turns, accelerations, altitude changes and maneuvering to keep on course are. Upon arrival in a terminal area, aircraft are placed in arrival queues and follow Standard Approach Routes (STARs). Aircraft are handed off between terminal area controllers and enroute controllers and between two enroute controllers at predetermined sector boundary points. This involves tying up the controllers' attention and use of communications channels. Minimum safe allowable spatial or time separations are defined between aircraft pairs, and search logic in the simulation detects projected or actual violations of the separation rules. The incidence of such violations, called conflicts, is considered most undesirable. When a potential conflict is detected the responsible controller takes action to prevent it from becoming an actual conflict. Such action may involve the delay (slowing or holding) or diversion (up, down, or sideways) of one or more aircraft. These actions also involve use of communications channels.

While the simulation is running, measures of system performance are calculated. These measures relate primarily to the capacity and safety of the system. They include the total and the geographically distributed flight load, controller workload, total communications usage, measures of system response time and margin of control capability, number of conflicts, etc. The system in action is portrayed on graphical displays in a form which is realistic and readily understandable. The user can choose displays of current system status to see areas of congestion or potential hazard, or he can display measures integrated over the run to get a feel of system effectiveness. Hard copy printouts are also available. A considerable effort has been expended to make the simulation useable by people without computer programming experience.

It can be guessed from the preceding that the simulation program is constructed in four parts: the data input package, the flight simulation, the ATC simulation, and the output package. Figure 1 is a diagram of the simulation.

The input package performs the functions of network entry, schedule entry, parameter entry, parameter changes, and interrupt control. Network entry consists of a fairly standard network editing program using teletype, and displays. Nodes representing terminals or VORTACs can be added to, moved, or deleted from a network. Links representing airways can be inserted to connect nodes. As a network is entered, it is immediately displayed on the scopes, and network data suitable for flight and ATC simulation is generated. This data is in the form of a 32 word block corresponding to each terminal or VORTAC node and containing the data for all the links to that node. A network may be saved away as a disk file and later recalled and further edited or for use in a simulation (i.e. a network file name is entered as a simulation parameter). The capacity of the level 1 model is 40 nodes and 240 links.

Schedules consist of a sequence of flights ordered by takeoff times (hour and second). The maximum period of simulation is 24 hours. Schedules are prepared offline, using a separate program, and are stored on disk. There is no practical limit to the number of flights a schedule file may contain. A schedule entry (see table 1) will contain a flight identifier, the scheduled departure time, the aircraft type (we have 5 classes: 4-engine jet, short haul jet, jumbo jet, SST, and prop general aviation aircraft) and the nodes comprising the flight route. A schedule file name is entered as a parameter whenever the simulation is started.

Airline Flight No.	Depart	A/C	From	Over		To
AL684	0700	0	ORD	CRL	CLE	PIT
EA567	0700	0	RDU	DCA	LGA	BOS
TW756	0700	1	PIT	CLE	CRL	ORD
TW738	0705	2	ORD	CRL	BUF	LGA
AA873	0710	1	DCA	LGA		
NW459	0715	0	PIT	FRR	DCA	
AL261	0723	0	ORD	CRL	BUF	

Table 1
Extract from a simulation schedule.

The parameter entry module is used to interactively enter parameters which are required in addition to network and schedule data. These parameters include aircraft characteristics, separation rules, time intervals required for controller functions and communications messages, and internal simulation parameters. This module can be used at the beginning of a simulation run or during a run upon interruption by a control routine. There are default values for all these parameters so the simulation can run without the module.

In the flight simulation, a timing control cycles through active flights in fast time. Fixed time steps are used instead of "event time" mainly to facilitate graphical output. The timing control also detects the presence of flights ready to depart and enters them into the simulation via a flight originate routine. Each active flight is represented by a 32 word dynamically allocated block containing its schedule entry and its current status, position, heading, velocity, altitude, assigned heading velocity, air altitude, etc. The Flight Update routine moves the aircraft through the network. During the enroute phase maneuvers for flight path correction will be needed occasionally but aircraft will usually be flying level along a straight line and thus position update can be computed very rapidly. Since many maneuvers are performed in the takeoff and arrival phases, a special Terminal Flight Update routine is employed for them. Arriving aircraft are placed in arrival queues and, after landing, a Flight Terminate routine removes the flight from the system, freeing its storage block. The Flight Update routine periodically (not on every cycle) calls a correction routine that imposes a weak control on the flights to keep them on the airways as they tend to wander off-course due to navigation and computer roundoff errors. The flight simulation can function independently of the ATC simulation, but aircraft may fly "through" each other or arrive at the same time.

The ATC Simulation imposes a simple air traffic control system upon the Flight Simulation. The purpose of this ATC Simulation is to ensure orderly and safe traffic in the simulated flight system. Orderliness is accomplished through the implicitly modeled flight tracking function and through the handoff procedures which take flights from takeoff through terminal airspace to enroute airspace to terminal airspace and landing. Safety considerations are introduced into the model by defining certain situations as inherently unsafe (as in the real ATC system). Procedures and control mechanisms are instituted to correct these unsafe conflict situations when they occur.

The ATC Simulation operates as follows: The Conflict Determination Routine interacts with the Flight Update and Terminal Flight Update routines to determine whether the extrapolated positions of active aircraft would result in conflicts. These extrapolated (potential) conflicts, as well as actual conflicts, are kept in the Conflict Lists. The ATC Process Supervisor picks up the potential conflicts (and handoffs as well), stores them in the Situation Queues and hands them to the Controller Functions as action situations to be evaluated. Corrective measures are instituted, if required, by the Communications and A.T.C.Hardware Module. This activity, in turn, alters the paths of aircraft in the Flight Simulation, thereby closing the ATC feedback loop.

The Conflict Determination routine tests at each time slice whether any aircraft in flight will be in conflict 5 minutes (a variable parameter) from that time by extrapolating its flight path. If the test is positive, two tasks are performed: 1. the details (aircraft numbers, jurisdictions, conflict type, detection time, and predicted conflict time) of the potential or actual conflict are added to the conflict list, and 2. the ATC Supervisor Process Routine is alerted for a controller action situation. The ATC Supervisor Process Routine, in turn, will act as a control box for the ATC actions and information flow required to prevent the potential conflict from becoming an actual conflict.

ATC controller functions are simulated by a computer program requiring no intervention on the part of the analyst who is running and observing the simulation. The model addresses itself to the gross functional aspects of the controller's activity as he endeavors to ensure safe, orderly, and expeditious traffic flow. However, the model does not contain the many detailed controller actions. These actions are aggregated into delays for purposes of the level 1 model. There are several sector controllers in the model, so that several handoff functions can be handled in parallel with several conflict determinations if the need arises.

The principal controller functions are handoff, conflict detection, and conflict resolution. A handoff is defined as a change in controller jurisdiction over a flight. Five types of handoff situations are considered: enroute sector boundary transition, enroute to terminal transition, terminal to tower control transition, tower control to terminal ATC transition, and terminal ATC to enroute transition.

The model represents a handoff by introducing a work interval during which the controller is considered busy and hence not available for any other functions. The model tries to make the controller free, and thus able to pay attention to the handoff, early enough to avoid holding of a flight. However, for various reasons, flight holding may still occur. Whenever this occurs the holding situation is recorded and later incorporated in the holding statistics.

The potential conflicts are detected by the conflict determination routine. When the simulated controller receives one of the four types of potential conflicts (overtaking, head-on, crossing, or altitude crossing) the conflict is added to the controller's situation queue (a list of all the current tasks the controller has to perform). The performance of each task entails a delay during which the controller is busy. Should the controller be busy with other tasks, he might not be alerted to the potential conflict in time to prevent an actual conflict from occurring.

When a request for the resolution of a threatening conflict appears at the head of a controller's situation queue a logic menu is consulted. This menu lists each type of conflict, type of conflict region (enroute or terminal), the actions required for resolution (slowing down, holding, altitude change, diversion), and delays due to controller decision and communications. After the appropriate delays, the prescribed actions are implemented by the pilots of the aircraft in the flight module. In the level 1 model conflict resolution consists of a single action but the mechanism exists for the addition of multi-action conflict resolution (e.g. climb one thousand feet and return to current altitude in five minutes).

The Communications Module models the messages encountered in air to ground, ground to air, and ground to ground ATC communications. The number of channels which the model provides to each controller is a variable (a typical number is 3 channels). Various types of messages are used in ATC functions and each message ties up a channel for a specified amount of time. This time represents a composite of message time, human factors, and acknowledgment time. For the duration of the message time the channel is tied up. Also, the action resulting from the message cannot take place until after acknowledgment.

The output package performs two functions. It calculates measures of system performance and it generates displays and printouts of these measures and of system status. The measures and statistics routine compiles the variables that are required to analyze the system performance. These variables fall into two categories: instantaneous values or measures that reflect the status of the model at a given time; and statistics which include integrated and average values and distributions. Statistics are continually generated, and may be obtained at any time by interrupting the simulation and calling for a display or printout of statistics or results. This action generates output for the period from the beginning of the simulation run to the point of interruption. After the printing of statistics, the simulation run may be continued. It should be noted that the simulation is not stochastic in nature - runs are exactly repeatable.

The measures and statistics found to be the most relevant are:

1) the number of controller action situations of each type (these include, of course, potential conflicts);
2) the total number of communications messages of each type;
3) the total number of actual conflicts occurring during the run;
4) System Response Time, which is the sum of the delay intervals between the inception of the potential conflict situation and the receipt by the pilot(s) of the ATC instructions that prevent an actual conflict from occurring;
5) Margin of Control Capability, which is the difference in time between the projected occurrence of the conflict had no ATC commands been issued, and the completion of the maneuver(s) by the aircraft in response to conflict resolution commands.

(1)-(3) may be divided by the total or maximum number of flights to get better comparisons between systems with different flight loads.

Output from the model is visible on two graphical displays. Hard copy output is produced by the line printer. On one display, a picture of the network is shown with appropriate moving symbols representing flights (see figure 2). The picture data representing the network is only computed when the network is entered, hence not much computer time is spent on this picture. The display of the moving flights must be recomputed every time positions are altered and this takes a fair amount of time. To save time the moving flights are only regenerated on every fourth time increment. The second display depends on the setting of the function switches. Three types of displays are available: (1) alphanumeric display of active flight identifiers, locations, status, speeds, etc. (this is essentially a numerical representation of the dynamic picture on the other display); (2) display of measures of flight load, including airways travelled, number of operations at each terminal, percentage of time communications channels are busy; and, (3) display of ATC load and safety measures, conflicts, controller action situations, system response time, and margin of control. The generation of the display files for these pictures takes a lot of time, so they are only updated on demand by the user. Printouts consist of snapshots of displayed alphanumeric data.

Resources

This section discusses the hardware, software, and manpower resources used in developing the level 1 ATS simulation.

The hardware on which the simulation runs is the Transportation Animated Graphics (TAG) System (see figure 3), consisting of:

- Honeywell DDP-516 CPU, 1 microsecond cycle time, 16 bit words, 16K core
- Vector General display subsystem, consisting of a display controller and two 21" CRT's, which can display 1000 medium length vectors and 1000 characters in 1/30 th sec.
- CDC 9433 moving head disk system, which can store 3 1/2 million characters with a mean access time of 75 sec.
- Honeywell 800 BPI Magnetic Tape Unit.
- Honeywell 300 lpm line printer.
- ASR 35 teletype.
- Data Channel (DMC) on the 516, which cycle steals and can transfer a word in 3.8 microsec.
- 16 Function Switches.
- RAND Tablet.

The application of minicomputers to large scale simulations has been viewed with justificable skepticism. We have found that for simulations where speed is not extremely critical a minicomputer with a large secondary storage capacity (such as that provided by our disk) has almost the power of a large machine. Lack of speed is no real problem in the case of a dedicated system like TAG. One must of course give some consideration to efficiency if long periods of time are to be simulated.

Connect times for interactive graphics tend to be long, thus, to keep costs at a reasonable level, one must timeshare a large computer or use a dedicated minicomputer. When the requirements of dynamic graphics are added, even the timeshared approach will necessitate a minicomputer to drive the displays. We feel that given a minicomputer for graphics it is reasonable to add a disk and more core and then try the entire simulation on the small machine. This relieves one of worrying about availability of and links to a larger machine.

Another advantage of the small dedicated system is that it is tractable. One can afford to keep on the project staff an engineer who knows almost everything about the hardware, who can fix the machine if it goes down, and who can design and implement interfaces for most vendors' peripherals. The latter means that one is not a captive buyer to the mainframe manufacturers' possibly expensive or inferior peripherals.

TAG Graphics equipment consists of a fairly standard high speed display controller and two scopes with different phosphors (expandable to eight). Pictures are drawn and refreshed from display buffers in the DDP-516 core and sent to the graphics via DMC. Refreshing slows down the 516, but not enough to be a problem in the level 1 simulation. Increased speed requirements of more advanced simulations could be satisfied by faster, 1 microsecond (DMA) data channel and a refresh core buffer as part of the display controller.

There are many reasons for using dynamic graphics in system simulation, with the best one in our case turning out to be ease of debugging. When one sees a picture of the network configuration, and the aircraft are behaving strangely, it is much easier to figure out what is going on and why than if one has to look at reams of printout. People skeptical as to the validity of the simulation or unfamiliar with technical details are more easily convinced if they see a picture of the simulated system.

Multi-scope graphics are highly desirable. Most of the cost of a display system is in the controller, thus, since one controller can drive several scopes, extra scopes can be added at a small cost. These scopes greatly increase the amount of information that can be displayed. For our simulation, we combine a dynamic pictorial display of the ATS network on one scope with alphanumeric displays of status data or integrated statistics on the second scope. The switch box allows us to select from a variety of displays for the second scope.

The line printer, while not a principal resource of the simulation, is worth mentioning. It is unfortunate that most small computer installations (including initially ours) do not have one, because a great deal of time is wasted in typing program listings on the teletype or in getting them printed on another computer. Our programming throughput was at least doubled after the printer was installed.

The Software configuration of the level 1 model is as follows:

1) a simulation program of 5.5K words with 3 1K overlays;
2) a 1K word graphics executive;
3) 7K words of dynamically allocable data storage, shared by the simulation and the graphics;
4) an independent 2K word program to enter schedules as disk files.

The programs are written in a macro language based on the DDP-516 assembly language. The macro processor is similar to SIMCMP of Orgass and Waite[4] but has added recursive features and is much more powerful than ordinary macro processors.

The reason that we did not use a higher level or a specially tailored simulation language is that no good higher level languages are available for our hardware. We also anticipated having to squeeze as much computing power as possible out of the 516, and to do that one must generally stay close to machine language. Although some flexibility in modifying programs may have been lost, programming is not too cumbersome in our low level language.

There are two interesting programming techniques used by the simulation that help extend computer capacity: overlaying and dynamic storage allocation. The routines comprising the simulation proper are always resident in core, but those used for data input, parameter updating, and output generation are overlayed. We had thought at first that overlaying would present a difficult problem, but it turned out that using our disk operating system and two routines to swap in and out of core, the overlays are executed simply and effectively. Our splitup of what is permanently resident in core and what is overlayed is a good one in that it does not affect the running speed of the simulation. Overlaying, done when the user stops the simulation to look at statistics, is fast enough so that no noticeable delay is encountered between the request for statistics and the appearance of these on the scope.

Storage allocation is perhaps the weakest link in our software. It is not that we do it inefficiently, but that we cannot seem to get around the principal limitation of the simulation: storage capacity which restricts us to 128 active flights. It would be very desirable to be able to overlay flight data and graphics data, but we are unable to do this without an unacceptable loss of operating speed.

The total amount of manpower allocated to our project was four man years, half of which was spent on hardware acquisition, interfacing, and systems software. Being novices to Air Transportation, we spent about one man year learning what ATC is, how it works, and how it can be modelled. Programming and debugging of the simulation took four months.

Experiments and Results

We have run the simulation with a standard network of 19 nodes representing an area bounded by Quebec and Raleigh, Boston and Chicago (figure 2). Input parameters were chosen to represent current day aircraft and current operation of the Air Traffic Control System. We imposed varying flight loads on the system with schedules derived indirectly from the Official Airline Guide. Our schedules had 250-450 flights between 0700 and 2200 hours, with reasonable diurnal and hourly departure distributions.

The simulation was able to handle these schedules with the following results (see table 2):

a) the number of handoffs increased linearly with flight load;
b) the number of potential conflicts increased faster than linearly with flight load, possibly quadratically;

Total Flights	250 flights	450 flights
Peak Instantaneous Airborne Count	38	66
Handoffs	631	1124
Potential Conflicts - Enroute		
Head On	10	44
Overtaking	6	56
Intersection	8	14
Altitude Crossing	5	11
Potential Conflicts - Terminal		
Head On	5	10
Overtaking	56	191
Intersection	45	139
Altitude Crossing	29	112
Potential Conflicts Total	164	577
Actual Conflicts	42	188
Communications Messages	1011	2084
Controller Actions	648	1450
Margin of Control Capability (sec)	125	22

Table 2

Simulation results for the 19 mode network with 250 and 450 flights for a period from 0700 hours to 2145 hours.

c) system response time increased slowly with flight load;
d) the margin of control capability decreased with flight load, somewhat erratically - for some high loads it became negative indicating an extremely unsafe situation;
e) controller workload was more due to handoffs than to conflicts, hence it increased slightly faster than linearly with flight load;
f) a certain fraction of conflicts were not resolved; this fraction remained constant with flight load;
g) the conflicts (see f) tended to be in the terminal area with at least one aircraft arriving;
h) eliminating the peaks of the departure distrib-

utions "on the hour" and "on the half hour" did not seem to improve capacity, ATC load, or safety.

The following conclusions can be drawn from our experience with the simulation:

a) the results of the runs, while not to be trusted quantitatively, all make sense, except for f) above;
b) simulation of complex systems is possible on minicomputers;
c) parts of our simulation could be used with minor changes for simulating other transportation systems;
d) conflict detection, which involves computations proportional to the square of the number of active vehicles does not slow down this simulation;
e) overlaying is remarkably easy with a disk;
f) core storage capacity and not speed seems to be the limiting factor in our hardware;
g) dynamic graphical displays are very useful in debugging simulation programs and in presenting the work to the customer.

Applications and Future Plans

While the level 1 simulation met its objective of showing the feasibility of a certain approach to ATS simulation, it is by no means adequate to provide results to real world problems. We now have a real world problem for our model. One of the missions at our Center is the selection of candidate concepts for future ATC systems. Such concepts might involve satellite based time of arrival surveillance and navigation, satellite communication links, automated strategic control*, airborne collision avoidance, etc. We will use our model to determine whether such concepts are necessary or whether the present ATC system or its planned extensions are capable of handling the demand from about 1980 on?

We will also use the simulation to compare the effectiveness of candidate ATC systems (mainly as measured by capacity and safety) and to generate tradeoff data for decision makers who can arrive at a rational selection after combining our data with demand and cost estimates.

To accomplish our tasks, we must upgrade our level 1 model and our simulation facility. The main limitation on the level 1 model was core storage. We plan to double the core on our DDP-516 to 32K words and we will add a 262K word high speed drum memory. With the extra core, we will have a better set of measures and more faithful representations of an ATS (like weather, multiple runways, strategic control, etc.). The drum memory will be primarily used to store network and flight data by fast overlaying. We expect the upgraded simulation to handle networks with up to 400 nodes and variable paths, and peak instantaneous airborne counts up to 8000. With the drastically higher flight loads and the use of secondary memory, conflict determination timing will become more critical, so flights will be sorted according to position. This will ensure more or less sequential access of flight data (which is required by the drum), and will cut down the number of two vehicle interactions that must be tested, thereby increasing operating speed. To reduce the overload on the CPU caused by the graphics, we plan to install a faster, 1 microsecond data channel and add a separate 4K word core refresh buffer to the display subsystem. With these improvements we are confident that we can have a valid model of one or more large chunks of the continental U.S. and Oceanic areas to provide concrete results to be used in the generation, selection, and evaluation of future ATS alternatives.

*Each flight is given, perhaps several hours in advance of takeoff, a 4 dimensional profile (time and space) to follow in order to minimize potential conflicts.

References

(1) "Report of the Department of Transportation Air Traffic Control Advisory Committee," Vol. I, Dec. 1969, which discusses ARTCC and other "third generation" ATC features.

(2) "Evaluation of Air Traffic Control Models and Simulations" by Higgins and Mpontsikaris is a critical catalogue of ATC models.

(3) "Air Traffic Control Simulation Model Exploratory Study," by C. W. Burlin, United Aircraft Corp. Research Lab Report H170592-1, AD700928, Dec. 1969. This work merits something of an exception of the critiques given of ATS models.

(4) "A Base for a Mobile Programming System," R. Orgass and W. Waite, CACM, VOL. 12, No. 9 (Sept. 1969), 507-510.

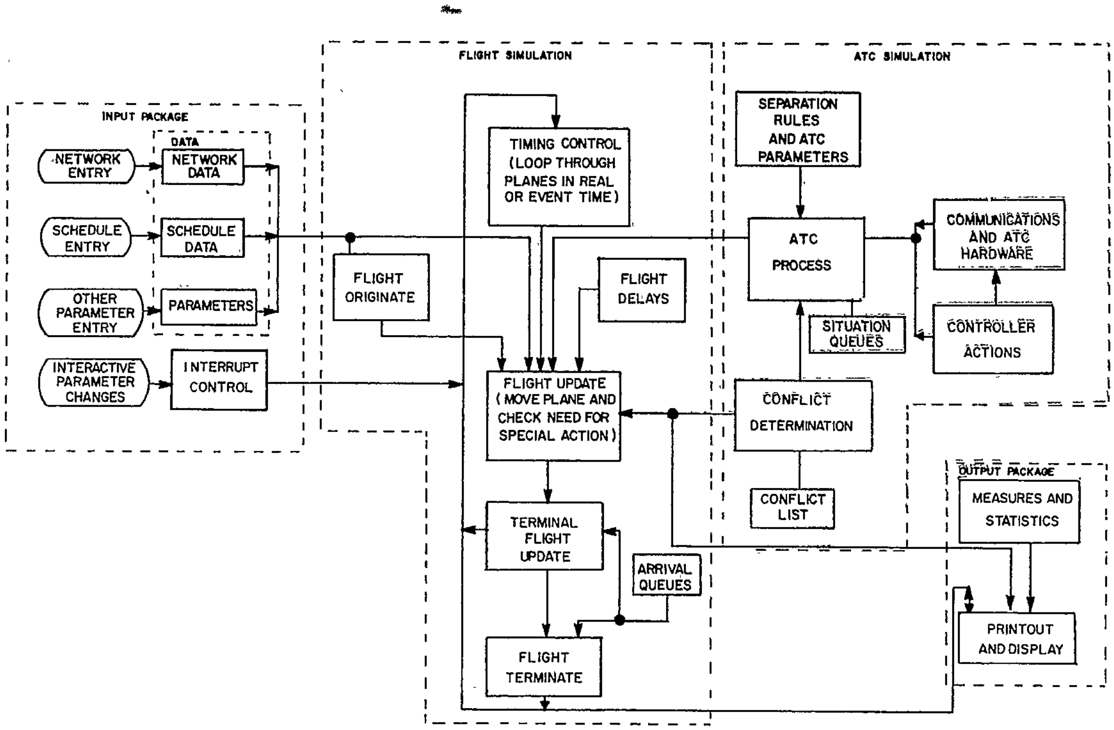

Figure 1.- Air Transportation System Network Model-Computer Program Outline

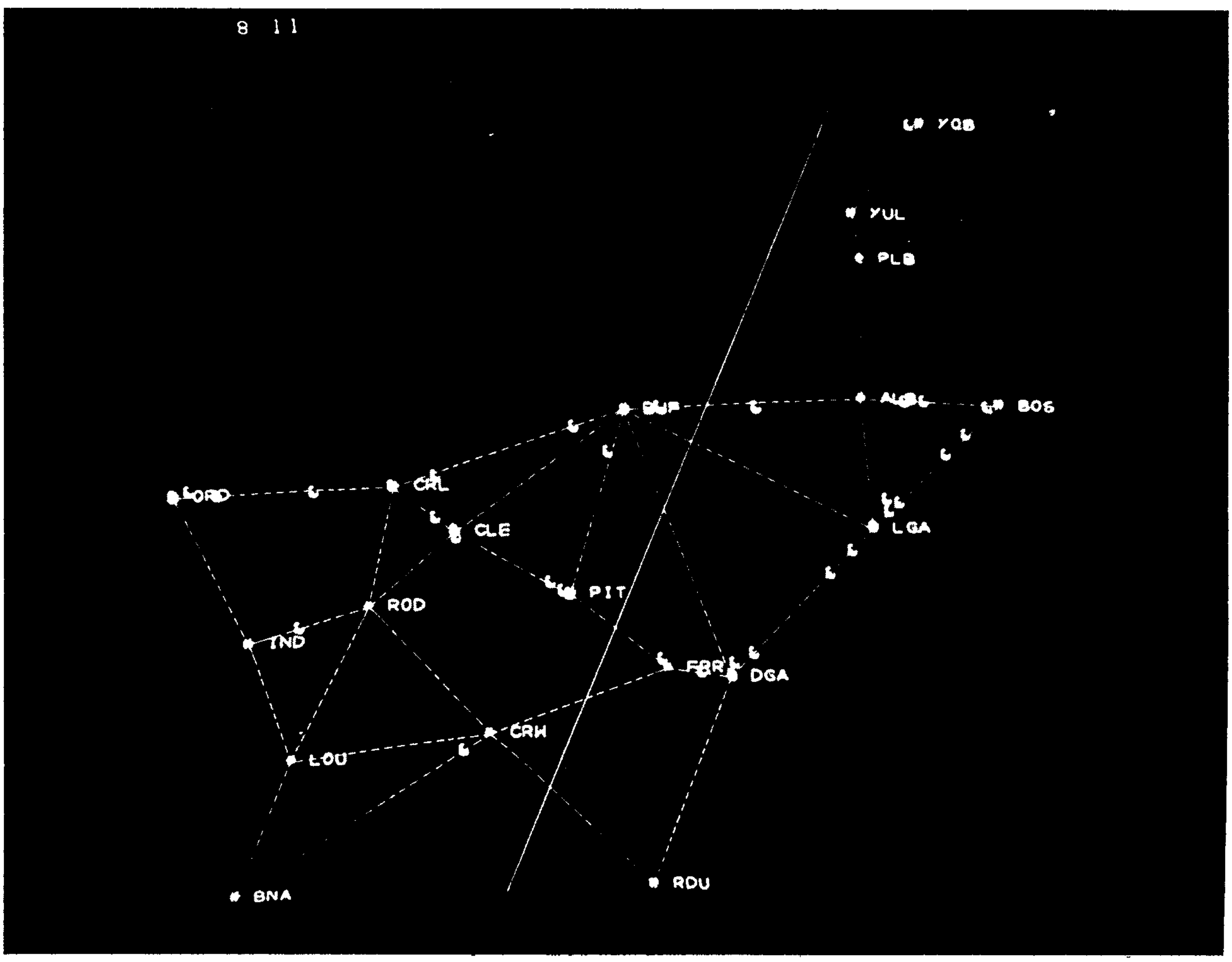

Figure 2

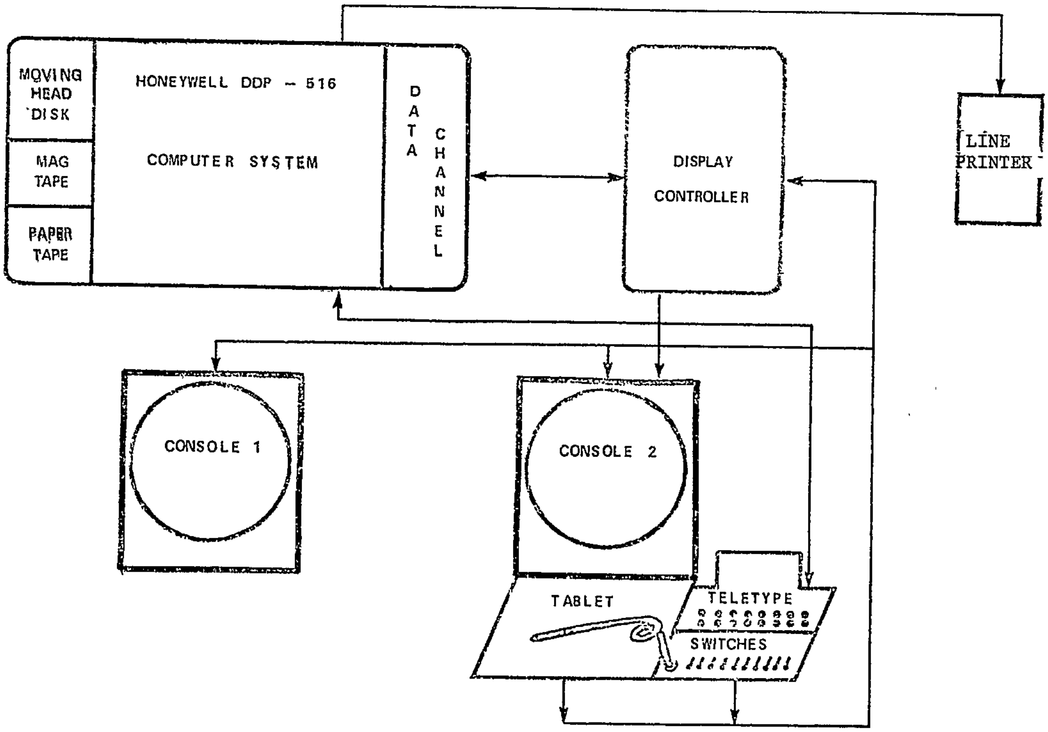

Figure 3.– Hardware Configuration