REPORT NO. DOT-TSC-FRA -71-4

MULTI-MODAL TRANSPORTATION SYSTEM SIMULATION

ROBERT C.RICCI
JEAN R.ROY
TRANSPORTATION SYSTEMS CENTER
55 BROADWAY
CAMBRIDGE,MA 02142

JULY 1971
TECHNICAL REPORT

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

1. Report No.
DOT-TSC-FRA-71-4

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
MULTI-MODAL TRANSPORTATION
SYSTEM SIMULATION

5. Report Date
July 1971

6. Performing Organization Code

7. Author(s)
Robert C. Ricci and Jean R. Roy

8. Performing Organization Report No.

9. Performing Organization Name and Address
Transportation Systems Center
55 Broadway
Cambridge, Ma 02142

10. Work Unit No.

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
Office of the Secretary
U.S. Department of Transportation
Washington, D. C. 20590

13. Type of Report and Period Covered
Technical Report

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A laboratory with real-time simulation capability is being developed for simulating the command and control functions related to transportation systems. The initial effort in Advanced Air Traffic Control Techniques is defining and evaluating the most effective role of controllers in future ATC systems. The present laboratory status, the simulation models and structure, and programming techniques that are being used are discussed.

17. Key Words

18. Distribution Statement

19. Security Classif. (of this report)
Unclassified

20. Security Classif. (of this page)
Unclassified

21. No. of Pages
24

22. Price

TABLE OF CONTENTS

LIST OF FIGURES

SECTION 1

INTRODUCTION

1.1 THEORY

This paper discusses our activity in Multi-Modal Transportation System Simulation which is being carried out at the DOT/Transportation Systems Center. A Multi-Modal Automation Laboratory (MMAL) with real-time capability is being developed for simulating the command and control functions related to transportation systems (e.g., Air Traffic Control (ATC), highway, and urban mass transportation). The first phase of this effort is producing an Advanced Air Traffic Control Simulation Laboratory to define and evaluate the most effective role of the controller in advanced generation ATC systems and to assess quantitatively various control strategies and automated techniques in these systems. The objective is to provide and exercise a system of interactive experiments to verify experimentally critical concepts of advanced ATC systems using fast-time and real-time computer simulations.

Gains in airspace capacity beyond those obtainable by improvements recommended by the Air Traffic Control Advisory Committee (ATCAC)[1] for the traffic of the 1980-1990's, and beyond, suggest a change in the basic method of control in order to reduce the control workload per aircraft. This change can be achieved by utilizing a strategic control concept where aircraft are assigned route time profiles.[2] Strategic control is referred to in the literature[3] as "a futuristic system in which aircraft follow route time profiles developed on the ground prior to the initiation of flight".

Several alternatives must exist for obtaining the benefits available from use of strategic control methods, since the system must accommodate a spectrum of airborne capabilities. Various alternative concepts, e.g., strategic, hybrid, or cooperative systems should be addressed and evaluated by means of computer simulation.

The present status of this laboratory and the particular computer simulation and programming techniques that are being used are discussed in this paper. Two different and unique approaches to the simulation have been developed and are described. Also, the role of analysis in computer simulation is identified.

This ATC oriented program is planned to extend over a period of several years. Since continuing investigations are expected to provide new ATC knowledge, the programs created for the simulation are being made extremely flexible and easy to expand and change. In addition, other transportation system problems will require real-and fast-time computer simulations. It is planned to use the same facility and many of the same computer programs to solve system problems for surface transportation systems, as well as for air transportation systems.

1.2 EXPERIMENTAL LABORATORY

The laboratory (Figure 1) consists of a major computation facility (a PDP-10 digital computer with 64K of 36-bit words and six million words of mass storage) and a display system. The PDP-10 system includes two magnetic tape units, eight DECTAPE units, a high-speed paper tape reader and punch, Calcomp plotter, ARDS display, line printer, high-speed INKTRONIK printer with an ASR33 keyboard, and an ASR35 console teletype. Five directly connected ASR33 teletypes and nine data-phone lines (110 baud) provide the interactive medium. The monitor we are currently using is a 10/50 multi-programming swapping system (DEC designation 5SO2) with real-time user capability and core-locking features. DEC's 5SO3 monitor with real-time queuing capability will be used when available.

The display system is a Honeywell DDP-516 computer with 32K of 16-bit words and an eight-million byte mass storage disk unit coupled to a Saunders Associates ADDS/900 display unit. The ADDS/900 display unit contains a display generator with hardware vector and character capability, light pen, functional keyboards, trackball, and data tablet. The DDP-516 includes a high-speed paper tape reader and punch and an ASR35 console teletype. To tie these two computers together, a

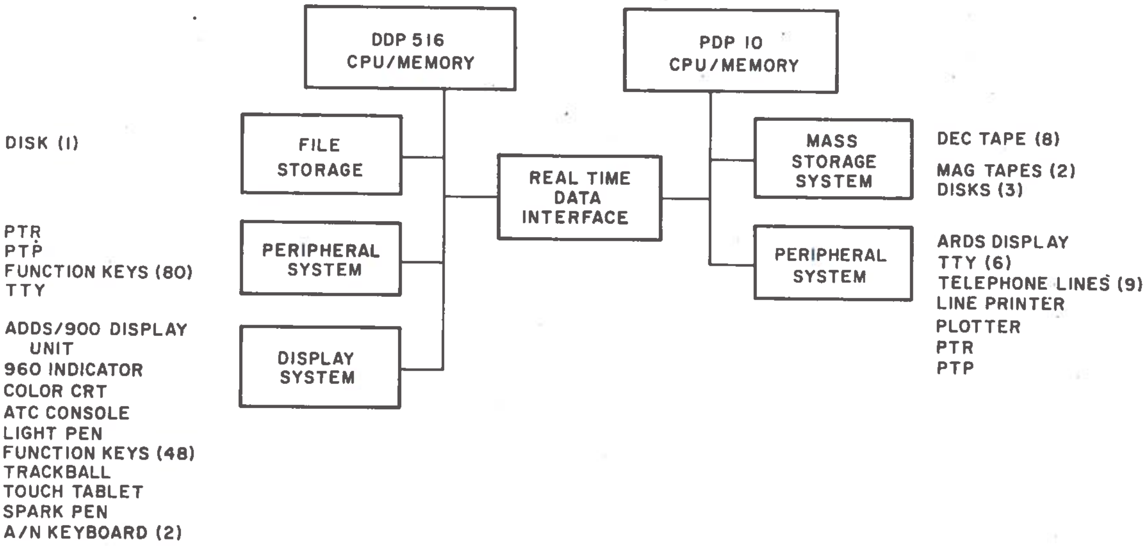

Figure 1. - ATC Laboratory

16-bit duplex data transfer interface designated RNS-10 was designed and built. The RNS-10 is connected to the I/0 bus of the PDP-10 and the DDP-516 with priority interrupts to both computers. All software linkages to the RNS-;0 can operate either under program or interrupt control.

One powerful mode of operation uses the PDP-10 as the ground-based date acquisition and command and control system with the DDp-516 computer as the display processor. Basically the functions that are performed by the PDP-10 computer are: (1) simulation of aircraft traffic including models for the pilot, aircraft, terminal area, and environment, (2) control of traffic with modular automation and decision making algorithms, (3) manipulation of data files and control to update the display list, (4) scenario generation for each simulation, (5) data recording and reduction, (6) controller/pilot aircraft command processing, and (7) flight strip printing. The functions performed by the display processor are: (1) generation of synthetic background video maps, (2) display file maintenance, (3) display refreshing, (4) I/0 handling for data entry devices, and (5) controller command execution.

In support of the above operation and between real-time runs the PDP-10 operates as a time-shared computer performing the following functions: (1) generation and debugging of all PDP-10 user programs for experiments, (2) post-run analysis of data generated by simulation runs, and (3) improvement of system software. The functions performed by the display system either with the PDP-10 or separately are: (1) generation, and (3) improvement of system software.

SECTION 2

SIMULATION

2.1 INTRODUCTION

The simulation model (Figure 2) has four basic areas: traffic generation, sky and terminal environment, pilot module, and control module. The traffic generator module contains the driving mechanism of the model and introduces flight characteristics limits, control response delays, and errors corresponding to the type of aircraft being simulated. It also introduces the update timing and sequencing characteristics of the data acquisition system and the data acquisition system errors. The present simulation contains data acquisition parameters which simulate the ASR terminal area radar system (primary and secondary radar).

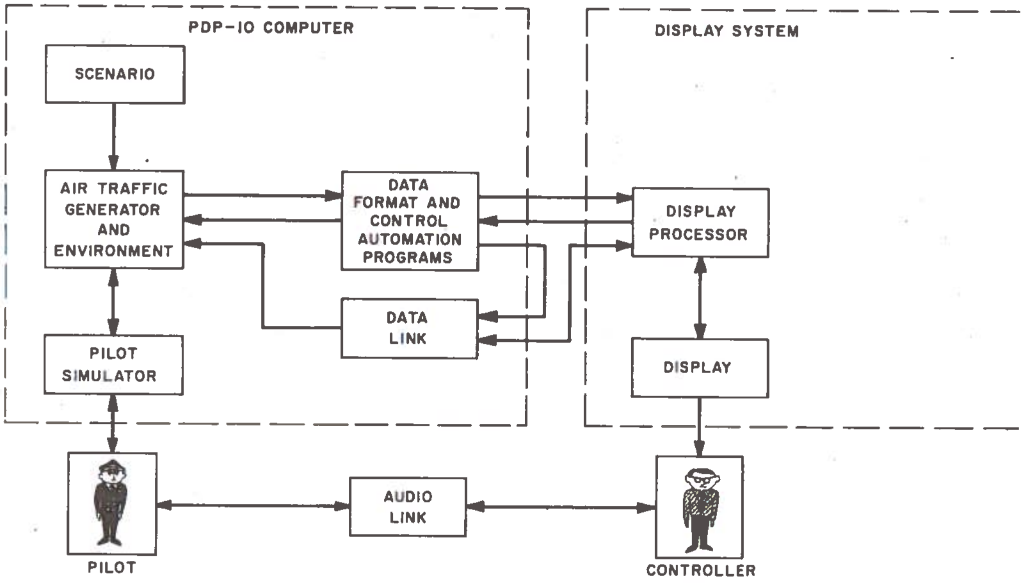

Figure 2.- ATC Simulation

A scenario generates aircraft arrivals and departures according to representative random schedules or actual traffic obtained from a terminal. The scenario provides a repeatable mechanism for comparative runs. It also initiates the simulation run and introduces all of the circumstances required to simulate special events such as airborne equipment failures, runway closures, and reversals, and missed approaches. A separate program provides for the semi-automatic generation of this scenario file in accordance with input data and statistical conditions related to the air traffic parameters.

The environment module includes flight characteristic limits, control response delays, aircraft errors, and a geographical section that contains position and route definitions. The pilot module accepts pilot commands for heading, speed, and altitude and then implements these commands to the aircraft (a/c). The controller is provided with a realistic display of traffic and can command a pilot through an audio link or a simulated data link. The input/output for the pilot operator is through an ASR33 teletype or an ARDS storage-tube display console directly to the PDP-10 computer. The controller may control display formats and presentation through various I/0 devices to the DDP-516 display processor and/or the ADDS/900 display generator. A data link from the controller to the a/c is accomplished from the display processor to the PDP-10 via the RNS-10 link. This particular assignment of model sections to the two computers is attractive for two reasons: (1) it represents closely the computing functions in an ATC terminal, and (2) it is a natural breakdown of work loads on the two computers to match their respective capabilities.

One interesting requirement for the Pilot/Controller interface is the use of an ATC language that models the communication and control language used by real world controllers and pilots. A complicating factor in the simulation lies in the interaction between the scenario and the pilot simulator. The scenario is operating "open loop" and pilot commands must correctly override scenario inputs for an aircraft. Also, recovery to scenario control when previous controller commands are cancelled requires careful implementation.

The intent of the pilot module is not to model a pilot but to allow one individual to handle multiple aircraft and interface the audio commands of the controller to the digital input required by the PDP-10 computer.

2.2 PROGRAM EXECUTION AND PROCESSING

The programming languages used are Fortran IV and MACRO-10 (the PDP-10 assembly language). The program structure is modular for independent check out and executed in a manner to permit modules to be added, changed, and removed without disturbing the execution of the unchanged modules. All input messages are processed independently of the source. The programs achieve a good balance between modularity and flexibility and are, therefore, easy to set up, change, execute, and evaluate. Maximum use of the PDP-10 software system and its features was attempted and the MACRO-10 assembler, Fortran IV compiler, loader, TECO, PEP, DDT, and Fudge programs were utilized.

The PDP-10 monitor system (10/50-5S02) is utilized to control and monitor the execution of the simulation program package. This monitor system includes both single-users-real-time and multiple-users-time-shared capabilities. It is possible for both capabilities to be used simultaneously, i.e., for one part of the program to be operating as a single-users-real-time program and to have the other parts of the program operating as multiple users on the time-shared monitor.

It is possible to run the simulation programs faster-than-real-time. It is recognized that the fast-time operation of the simulation precludes the interaction of the controllers and pilot station operators on any meaningful basis. The fast-time mode is expected to be useful, however, to review simulations prior to real-time interactive experiments, for completely automated experiments, and for sensitivity analysis studies.

Wherever feasible, the programs are written to minimize PDP-10 processing time. Unnecessary arithmetic processing such as frequent periodic updating of aircraft positions without regard for the operation of the simulated data acquisition system is avoided. Only those positions corresponding to simulated tracking program outputs based upon simulated data acquisition system input data are calculated. The complexity of the simulation programs is minimized and is sufficient to establish a valid basis for examining the man/machine relationship between the air traffic controller and the facilities provided to permit him to execute control over his assigned traffic. Suitable characteristics of aircraft flight which do not affect the appearance of the controller's situation display or the content of other data provided to him are not included within the simulation processing. Similarly, errors associated with the aircraft responses to control commands and with the inaccuracies of the data acquisition and processing system being simulated, which are too small to be noticeable to the controller, are not included within the simulation processing.

2.3 PROGRAM IMPLEMENTATION

2.3.1 Introduction

Two different and unique approaches to the simulation have been used resulting in two different simulation packages for the same problem. This will allow cross-checking of results for particular cases. Each has its own characteristics; one emphasizing real-time operations, and the other faster-than-real-time operation with automated control of aircraft. A comparison of the features of each approach to meet the performance requirements for the simulation is shown in Table 1. These two approaches are described in detail in the following sections.

TABLE 1. - SIMULATION FEATURES

MACRO APPROACH	FORTRAN APPROACH	FEATURES
MACRO-10	FORTRAN-IV	MAIN SIMULATION LANGUAGE
"MAIN" PROGRAM	TIME-SHARING	SIMULATION CONTROL AND EXECUTION
COMMON DATA TABLES	HI-SEGMENT COMMON AREA	MODULE-SHARED DATA
LOADER "SYMBOL TABLE"	DEFINED IN "PILOT" USER JOB	PILOT DICTIONARY
MACRO EXPANSIONS	FORTRAN DEFINITIONS	PILOT LANGUAGE DEFINITIONS
INDIVIDUAL "REL" FILES	SEPARATE USER JOBS	MODULARITY
ASCII TEXT	ASCII TEXT	INPUT AND OUTPUT DATA
YES	YES	CONTROLLER/PILOT LANGUAGE
YES	YES	DATA TABLES INDEPENDENT OF MODULES USING DATA
YES	YES	ON-LINE "PILOT" AND/OR "SCENARIO" INPUTS

2.3.2 Macro Approach

In this approach[4], the air traffic simulation system is written in MACRO-10 assembly language. One implementation goal was to make maximum use of existing software capabilities in order to minimize the amount of implementation effort required. To this end, the preplanned scenario input files and data recording output files are formatted as ASCII test. System inputs and outputs of this kind are, therefore, in a form readable by both the computer and humans.

Capabilities for defining problem specific data such as wind conditions, aircraft performance parameters, navigation fix positions, and aircraft routing and control procedures are provided through MACRO features of the PDP-10 assembler. This approach provides a standard series of MACRO definitions which implement simple but problem-specific languages for describing air-traffic-control-related data items. Source text files containing both MACRO definitions and user defined specific data are assembled together to make relocatable modules (REL) for use in the simulator. In this fashion, convenient language forms are provided to the user for defining problem-specific data required for the simulation.

One consequence of this approach is system modularity. A specific simulation system is created by loading together a number of (REL) modules, each of which performs a particular function for the total simulation. Some of these modules are relatively permanent, while others are intended to be extremely flexible and variable. Service functions such as input/output, display generation, and aircraft motion should be stable. The modules for these functions should change infrequently. Other ATC related modules which contain wind profiles, navigation fix positions, routing procedures, and the like will change often and will be created specifically for a new simulation experiment.

The standard PDP-10 loader is utilized to link together the system modules into an executable simulation program. The symbol table created by the loader during the loading process constitutes a valuable resource and is utilized as a symbolic dictionary by the simulation program at run time.

Another important feature of the air traffic simulator is that all linkages and data accessing mechanisms operate strictly on a name basis. Data references to specific components of the aircraft data table are by data element names. The organization and order of data elements within the aircraft data table is irrelevent to the programs which use these data. In fact, the data table can and has been rearranged several times without any changes in the other simulation programs. Accessing functions for obtaining and inserting data into this table are associated with the table and modularly separated from other simulation routines. This centralizes the management of the aircraft table data and permits flexible and convenient expansion in the future.

The simulation system is structured as a number of independent modules which, when loaded together, comprise an executable simulation system. Many versions of aircraft parameters, wind, routing, and recording modules are expected to be created as the relocatable files used as components are changed. The runnable simulation system created by the loader can be saved for execution at any later time.

The simulation system, when running, takes its input either from on-line pilot inputs or from a stored scenario file located on the disk. This scenario file is in a text format identical (except for timing indications) with the on-line input and can be created either by a special scenario generation program or by use of the PDP-10 text editor, TECO.

The operation of the air traffic simulator shown in Figure 3 can be conveniently broken down into the basic phases. Phase I involves the input of commands to the system from either the on-line pilot station or the stored scenario file. The input processing operations accept the text input data and convert it to an internal form. This output of the input processing is placed on a command stack associated with the appropriate aircraft in the simulation.

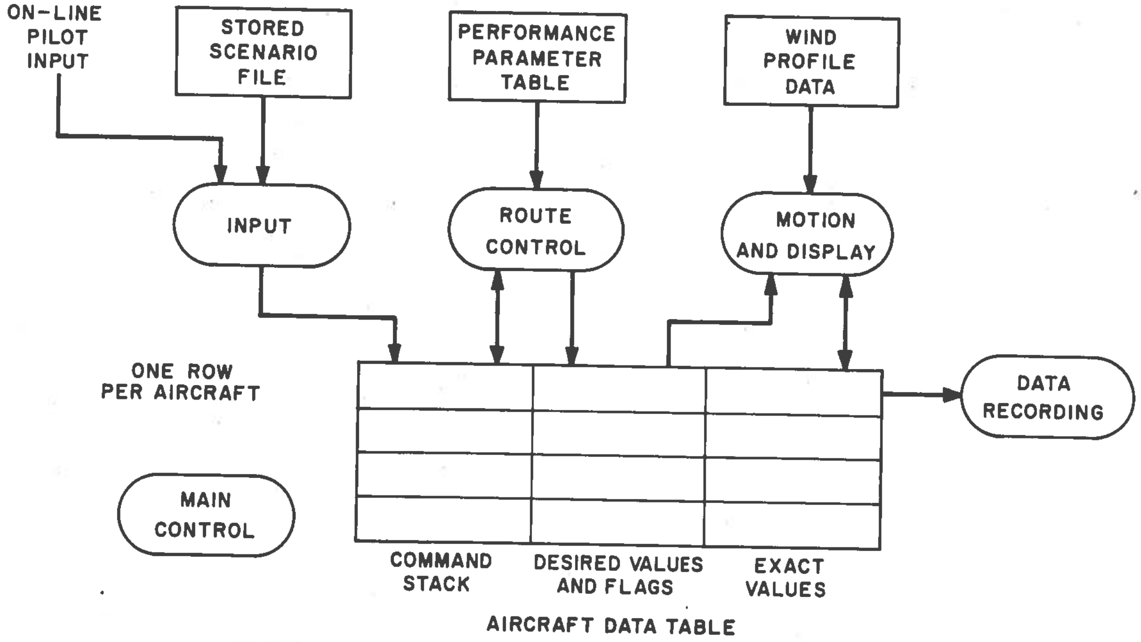

Figure 3. - Conceptual Organization of Simulator

Phase 2 involves the operations of route control which perform the strategic control of aircraft and deal with the navigation and position checking function normally performed by a pilot. The route control operations interpret the commands on the stack for each aircraft and determine an appropriate course of action. This course of action is reflected in particular desired values of altitude, heading, and speed which are assigned to an aircraft. The operations of the route control module are terminated with the proper assignment of desired aircraft values.

The third phase in the simulation involves the determination of aircraft motion and the appropriate display of the results. This module actually flies the aircraft in accordance with the appropriate dynamics, and updates its position and velocity vectors. The motion routines compare the exact aircraft values and desired values to determine if any changes in speed, altitude, or heading are required to conform with the desired performance. If so, the changes are effected before the aircraft is moved. The display functions use the data generated as a result of the motion operations and create a visual representation of the simulation.

A fourth appended operation of data recording is required to instrument the simulation and to determine the performance parameters of interest.

The entire simulation operates under the control of the main control module. This module has responsibility for calling the phases in the proper order, for determining the timing of the simulation, and at the start of the run, for the proper initialization of the system.

The other modules shown in Figure 3 represent data contained in the stored scenario file, the aircraft performance parameter table, the wind data table, and the aircraft data table. The sections which follow discuss the operation of the most important modules contained in the simulation.

Aircraft Data Table. - This module contains three elements associated with aircraft data: (1) the storage space required for each individual aircraft, (2) the service routines needed for dynamically allocated memory space when aircraft are created and for reclaiming that space when aircraft are deleted, and (3) a set of accessing functions which provide a symbolic means of assigning and retrieving data values for individual aircraft. These accessing functions are deliberately located in the aircraft data table module to isolate them from the other simulation system modules which will reference them symbolically.

A valuable by-product of the accessing scheme is the ability to incorporate an arbitrary subroutine to calculate some value not actually stored in the aircraft data table. From the external simulation system such a calculated value is indistinguishable from one actually stored.

Aircraft Parameters. - This module contains the definition of aircraft performance parameters. The implementation philosophy is similar to that of the aircraft data table. The major differences are that storing into this table is never permitted, and accessing is based on aircraft type name rather than through a handle.

The data values obtained by other simulation modules from this aircraft parameter module are calculated each time that an access request is made to produce a random value with the appropriate statistical properties. This mechanism provides a controlled degree of variability in the performance of aircraft. Each parameter is chosen from a set of Gaussian variables with the indicated mean and standard deviation. This accessing mechanism is more complicated than the simple table look-up used in the aircraft data table module, but externally it appears exactly the same. A default standard aircraft, is provided so that aircraft inadvertently made without an explicit type, nonetheless, refer to a set of performance parameters. This is required so that other simulation modules such as aircraft motion will continue to operate correctly. Nominal values are used for this default standard aircraft.

Motion Update. - The motion update module moves each active aircraft through the sky and adjusts its motion to conform with the desired performance values associated with that aircraft. In the update cycle for each aircraft, the first item performed is a call upon the wind module to calculate the wind velocities at the altitude of this aircraft. The next step is the altitude updating. The rate of climb is obtained from the aircraft data block, and the exact altitude of the aircraft is calculated. If the altitude change overshoots the desired altitude, the exact altitude is set equal to the desired altitude. The next portion of the update routine checks the desired speed and the exact speed of the aircraft to see if an acceleration or deceleration is required. If so, a speed change is calculated and the exact speed determined. As in altitude update, the speed change is checked for an overshoot condition, at which point, the desired speed and exact speed are set equal.

The next and most complicated section of the update routine involves correcting the heading of the aircraft. It first tests to see if the aircraft is navigating in the homing mode. If so, a check is made to see if the aircraft is within approximately two miles of its homing position goal. If it is, the aircraft is changed from homing mode to heading mode of navigation. If the aircraft is far from the homing position, the heading from the aircraft to that position is computed. To account for the effects of wind, a calculation of the drift angle which would be achieved if the aircraft flew on the heading to the homing position is made. This drift angle is then applied as a correction to calculate a corrected desired heading to compensate for the wind effects.

The exact aircraft heading is then changed according to the aircraft rate of turn to achieve the desired heading. The number of degrees to turn this update cycle is calculated and then the direction, either best way, left, or right, is determined.

The final portion of the update routine moves the aircraft along in space. The newly calculated exact heading and speed are utilized to calculate aircraft X and Y motion components. The effects of wind are added in, and the actual distances moved in X and Y are added to the exact aircraft positions. The new positions are stored back in the aircraft data table.

Routing Definition Module. - This module contains the definitions of air traffic control routes and navigation fix positions, as well as program routines and MACRO definitions required for operation of the route following procedures.

MACRO definitions determine the language forms available to the simulation system user for defining positions and aircraft routing procedures. An important feature of the user language available is the ability to combine several simple aircraft procedure steps into a more complicated step which can be given a name and called with parameters. One defines the name for the complex function and indicates the parameters which are to be used for input. Following this initial declaration, a series of steps comprising the actions of the complex procedure are set forth. These steps refer to other procedures and will call them with indicated parameters.

Display Output Modules. - Two separate display output modules are used. One module transmits display data over the RNS-10 link to the DDP-516/ADDS-900, while another handles display output functions on the ARDS display. Separate versions of the system can be prepared for use with these two displays or specified at run time.

In the first module, a DDP-516 output is concerned only with transmitting aircraft update information over the link; display files containing map data and aircraft entities, together with update functions, are handled by the Display Software system and additional code in the 516. Aircraft update information is transmitted in six-word blocks having the following format: Aircraft Number, X Position (scope units), Y Position (scope units), Altitude (100's of feet), Speed (knots), and Flight Number.

The second module handles display output to the ARDS. It treats the ARDS as an independent output device, leaving the controlling teletype free for command input and message output. It draws the background map lines and labels from tables. The tables contain pointers to the locations where numerical coordinates are stored, and in setting up the tables these locations can be called out by name. This allows reference, by name, to the geography points defined in the routing module.

The module plots a point on the ARDS for each active aircraft, once per cycle. First it picks up the X and Y position from the aircraft data table and does a set point to this location on the ARDS. It performs coordinate conversion and draws a unit length short vector to represent the aircraft on the scope. This display is not intended to replace the DDP-516/ADDS-900 but only to provide a device to allow the users to exercise the simulation in the PDP-10 when the DDP-516 is otherwise occupied.

2.3.3 Fortran Approach

As in the first approach, a modular program structure has been developed that provides for flexibility of running experiments and for future expansion.[5] Fortran IV is attractive because it is familiar to most programmers; it produces programs that can be run on a large variety of different computers, and it is structured such that program listings can be readily understood without a great deal of supplementary documentation.

In any program operating in real-time, provision must be made for some events to occur according to a schedule (e.g., when aircraft are introduced into the system, or when an aircraft has its position coordinates updated every radar scan). This implies that all programs processing data connected with some scheduled event must be run on time. This approach makes use of the time-sharing monitor to carry out these functions. Its "scheduler" allocates each user a time slice. At the end of this time interval, the user's program is swapped out of core, if required, and a new user is brought in and executed for his time slice. By breaking the program into logically distinct parts and making each part a separate user, the time-sharing monitor can be used to allocate time and to sequence users so all necessary processing can occur. To be assured that a scheduled function happens at a time, t, a user need only interrogate the system clock and perform that event during the first system allocated time slice during or after time, t.

The real-time computer program is divided into separable, major simulation functions as shown in Figure 4, scenario input control, pilot input control,^{6} aircraft data generator, aircraft reporting and environmental comparator, display data file and control, and clock control. These functions have been designed as separate time-sharing "users". Since in the normal time-sharing mode users are kept separate from each other, it is necessary to set up a shared data area through which the users can communicate. In order to optimize computer usage when the simulation is run in this manner, each user program is structured as shown in Figure 5.

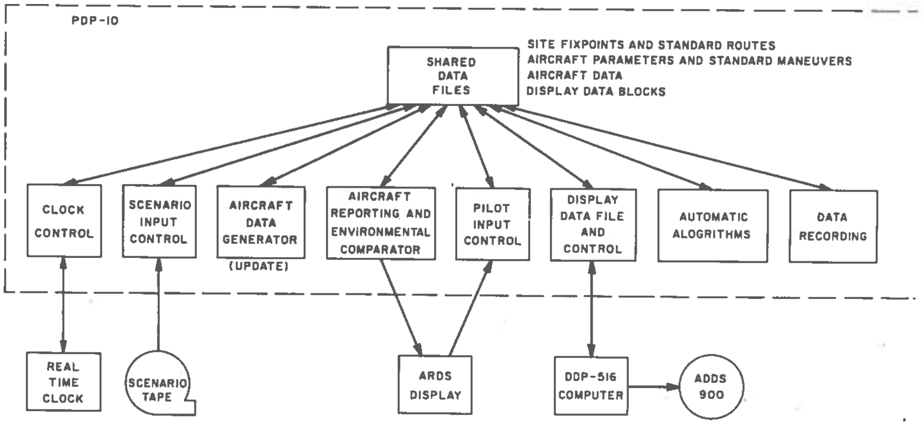

Figure 4. - Real-Time Program Structure

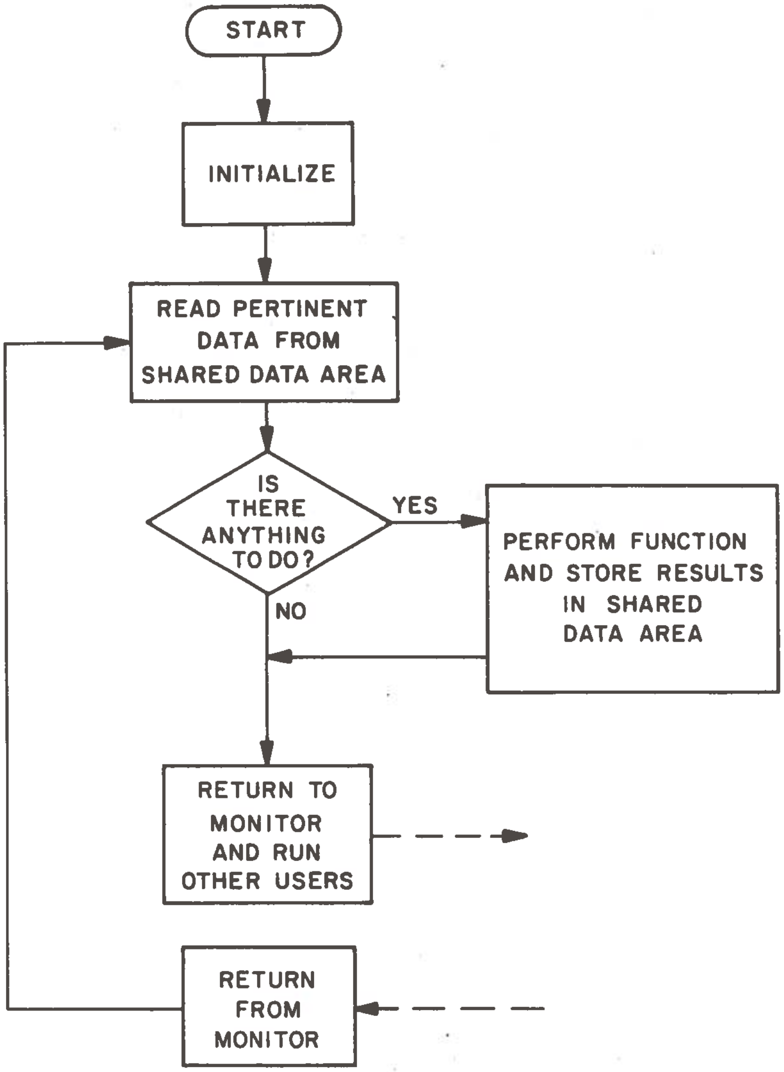

Figure 5.- Structure of Each User

After initialization, the user program reads the pertinent data from the shared data area. These data, for example, may be time or an update or start flag. Based on this information, the program then decides whether the related function should be run. If yes, the function is run and the results are stored in the shared data area. If no, control is passed to the monitor and the other users are run. After the other users are run, control is then returned to the first user. To run the simulation all users must be loaded into the PDP-10 and started, but the order in which they are loaded and started does not matter. The time-sharing monitor will continually cycle through the users causing time and therefore the pertinent data to be updated.

To implement the shared data area, a file is maintained with all symbols that are of interest to more than one user. This file is defined by FORTRAN COMMON statements with COMMON statements being compiled with each user's program. In addition, they are compiled into a block data program and saved in a file called COMMON.SHR. As each user begins execution, he links his program with this COMMON.SHR file. He can then write information into this file for use by any other user and can search the file for information left there by any other user. This file resides in a privileged area called the high segment which is an area in core that can be shared by many users.

This approach allows almost all of the programs to be written in FORTRAN. The modularity of this approach allows new functions to be added or old ones modified without disturbing working programs so that someone adding new features need only understand COMMON and his particular task. The simulation system may be tailored at load time by choosing the proper available users. The problem of providing and maintaining a master co-ordinating program is completely avoided.

The most important modules of the simulation system program package are: aircraft data generator, scenario input control, aircraft reporting and environmental comparator, pilot input control, and display data file and control. Although these modules are different in details of implementation, they are similar in function to those discussed in the MACRO approach.

2.3.4 Display Software.

The display parameter tables, data files, and associated updating programs are generated by a Dynamic Software System (DISS)[7], a higher order user's language and software system specifically developed for air/ground avionic display applications.

The features of DISS are: (1) simple command structure and format, (2) static graphics immediately displayed, (3) simple FORTRAN-like expressions to link mathematical model outputs to display parameters, (4) off-line operation to create, edit, and modify displays, and (5) a run mode in which a real-time environment is provided with display update and linkage for the total system.

With the Dynamic Display Software System, the user can interactively create the geometric characteristics of his desired display and specify the dynamic linkages with a program to simulate the display environment which resides in the DDP-516 or the PDP-10. After the desired set of indicators is specified, the system serves as a real-time simulator to evaluate the usefulness of the displays. When the system runs on-line with the PDP-10, a more complicated simulation environment is possible than with the display processor itself. The user has the capability of making changes to previously created indicators to provide an evolutionary means of developing environment indicator systems.

The system is divided into two large functional parts and one smaller part to allow efficient use of core. These parts are the create/edit phase, where display lists are created and matched via dynamic expressions to the outputs of a simulation; the simulation phase, where display and update of the data structure and operation of the simulation are under control of the real time monitor, and the input/output phase, which supervises transfer of display files between the create/edit and simulation phases, and changes all user entered specifications of dynamics into executable object code via a small compiler subprogram.

The first phase of system operation, the create/edit phase, is shown functionally in Figure 6. This phase consists of: (a) a user language, which provides the interface between the user and the system, and preliminary conversion of alphanumeric input into display elements, (b) data structuring routines, which handle storage allocation, creation, and maintenance of the data structure, and (c) connector programs, which handle overlay of the other system phases upon user request.

Graphic components specified via the user language are displayed immediately. Also specified via the user language, are dynamic expressions that modify the graphic components using the update program in the simulation phase. The language, however, does not generate object code for dynamic expressions but stores the expressions as packed characters for later processing by the mini-compiler in the input/output phase.

The input/output phase of operation services input and output of the data structure onto paper tape or disc. Prior to saving the data structure, a mini-compiler using pointers internal to the data structure compiles each dynamic expression of FORTRAN-like source code entered during the immediately proceeding create/edit phase. Once saved, the data structure may be loaded back into create/edit phase for additions or deletions or into simulation phase.

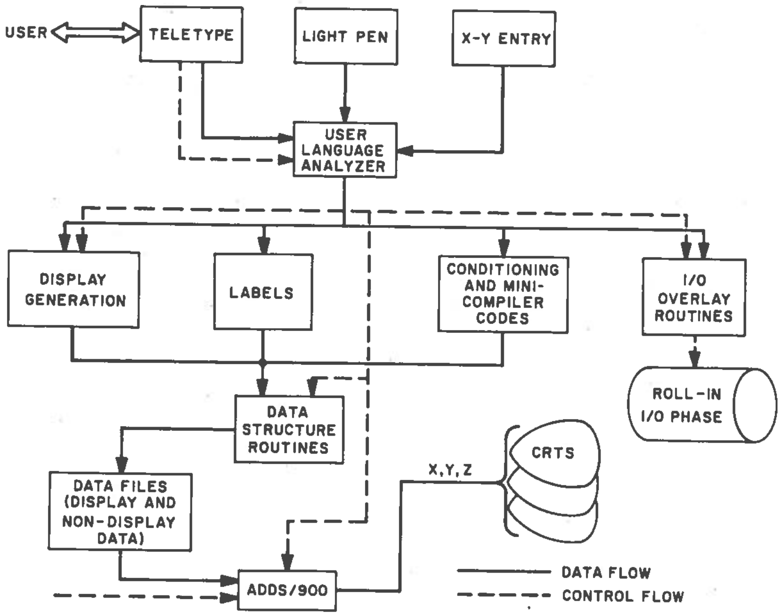

Figure 6. - Functional Design of DISS Create/Edit Phase

The simulation phase (Figure 7) provides an environment whereby the displayed data may operate dynamically. Provision is made, under user option, for input from or output to any external device. Inputs may be used directly as dynamic parameters, or they may be used by a model and/or background program which may alter predefined registers. The update program uses these data and mini-compiler code to make changes to the display list. It is at this point in the display system that data provided by a simulation in the PDP-10 are used to alter the display. In the stand-alone mode, the user generates a math model and a background program to replace in a very simple way the PDP-10 simulation package. This allows checkout of the simulation program of the display system.

A real-time monitor supervises the scheduling of the various routines and allows the user to modify the update cycle time.

The display data format is a hierarchical ring structure with auxiliary blocks which describe the parameters to be used for dynamic updating of the data. The highest level in the hierarchy is the frame ring which is a ring of frame blocks. Attached to a frame block is an indicator ring and an auxiliary block for register definition. Each indicator block has an entity ring associated with it, with each entity block having components and conditioning blocks attached.

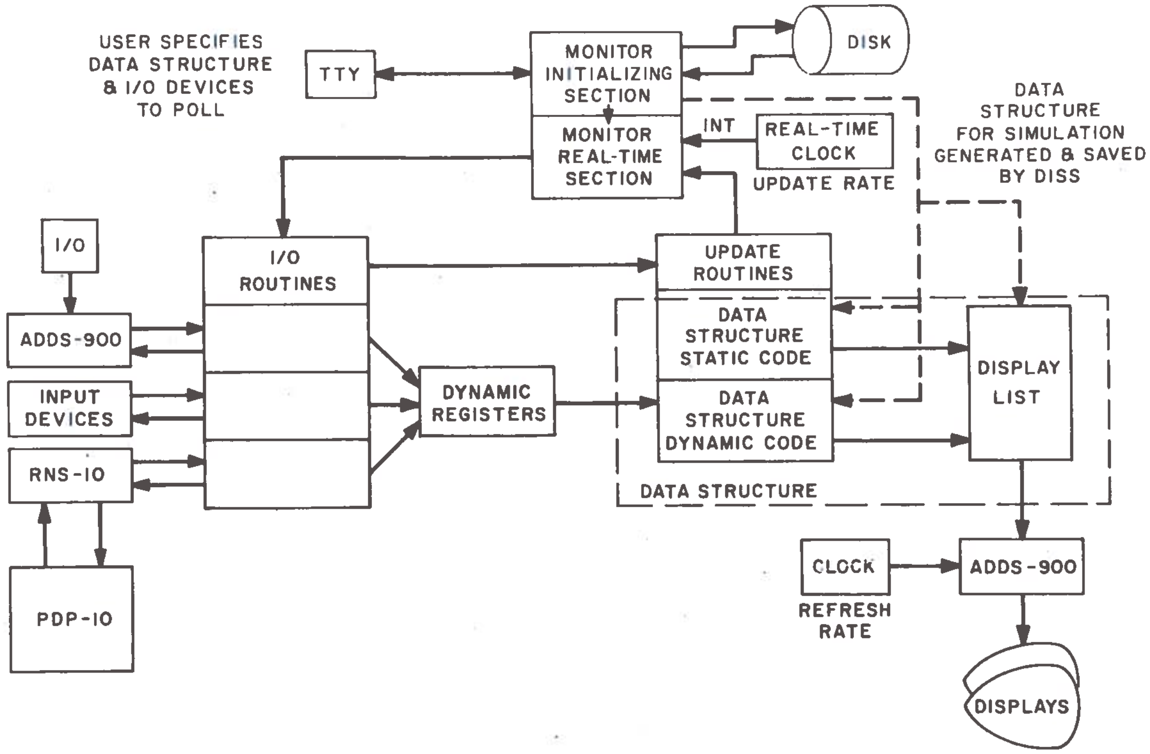

Figure 7. - DISS Real-Time Simulation Phase

The following is an example of how DISS generates a dynamic display from a single user language. The display (Figure 8) is an air-speed indicator and altimeter from a simplistic aircraft cockpit.

The indicator pointer and altimeter numerics are dynamic and are moved by a math model program which is not shown. There is one "frame" composed of two "indicators" with numerous entities within. The actual program typed in by a user is listed as follows:

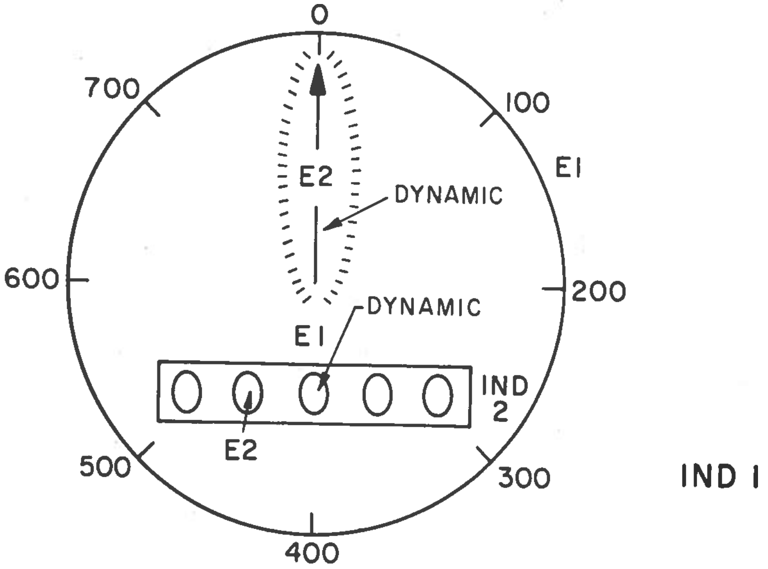

Figure 8. - DISS Example

```
:F 1 [Define Frame 1]
:DRE R10, 650 [Declare Registers Containing Initial Data]
:DRE R11, 25
:I1 [Indicator 1:  Air Speed Indicator]
:E1 [Entity 1] [Arc, Tick Marks]
:S 0, 250 [Position Point]
:A 200 [Draw Arc]
:TMA 0, 192, 8, 6 [Draw Tick Marks]
:SA 0, 200, 0, 700, 8, 1, 10 [Draw Scale]
:E2 [Pointer to Speed]
:S 0, 250
:R 0, 250, R8* (360/800) [Dynamic Conditioning, R8=Current
                          Air Speed, Changed by Model]
:L 0, 180
:L 5,-10
:L-10, 0
:L5, 10
:G R8, R10, 1 [Blink if Air Speed Exceeds 650]
:SK 1
:B
:I 1
:STI S, 3
```

```
:I 2 [Indicator 2:  Altimeter]
 E 1 [Altimeter Rectangular Boundary]
:S -75, 150
:L 150, 0
:L 0, 40
:L -150, 0
:L 0, -40
:E 2 [Altimeter Read-out]
:S -70, 160
:DR R3, 5, 2 [R3=Distance from Ground Zero]
:STI S
```

The program is now saved on the disk (by I/O phase) and used with the real-time monitor.

As the program is being generated in the create/edit phase, the actual picture is displayed so that corrections can readily be made.

2.3.5 The Role Of Analysis

The synthesis of control strategies for a terminal and/or enroute air traffic control system requires analysis to acquire inherent knowledge of system behavior. It is this knowledge at the present time that is lacking for the various control system requirements. In general, there exists an exceptionally strong requirement for analysis, synthesis, and simulation techniques that deal with those system and subsystem design parameters, variables, and characteristics that have actual operational applicability. The functional system shown in Figure 9 is presented as a reference ATC system. Given that definitions are applied to each function (except the scheduling and control strategy), both a scheduling strategy and control strategy must then be generated that can provide some acceptable measure of system performance. It is well within the realm of simulation technology to define the physical environment as a single terminal area or a "nest" of airports that represents a metroplex type of terminal area or even the national airspace with its several hundreds of terminal areas. However, the system may be defined, the control system design engineer must devise a control strategy for the spectrum of terminal area configurations and associated characteristics, e.g., weather, air routes, aircraft demand, etc.

Specifically, the dominant system variables and parameters must be determined. The functional and quantitative relationships among the characteristics of the data link, the physical environment, the aircraft environment and system performance in terms of terminal area I/O rates (e.g., arrival and departure rates), number and frequency of missed approaches, and number and frequency of potential collisions require examination. These types of sensitivity data and the statistical behavioral characteristics must be realized prior to the synthesis of control strategies. Significant analysis techniques can be brought to bear to acquire this information in a methodical manner as well as to formaulate preliminary control strategies as a direction function of these system and subsystem behavioral characteristics.

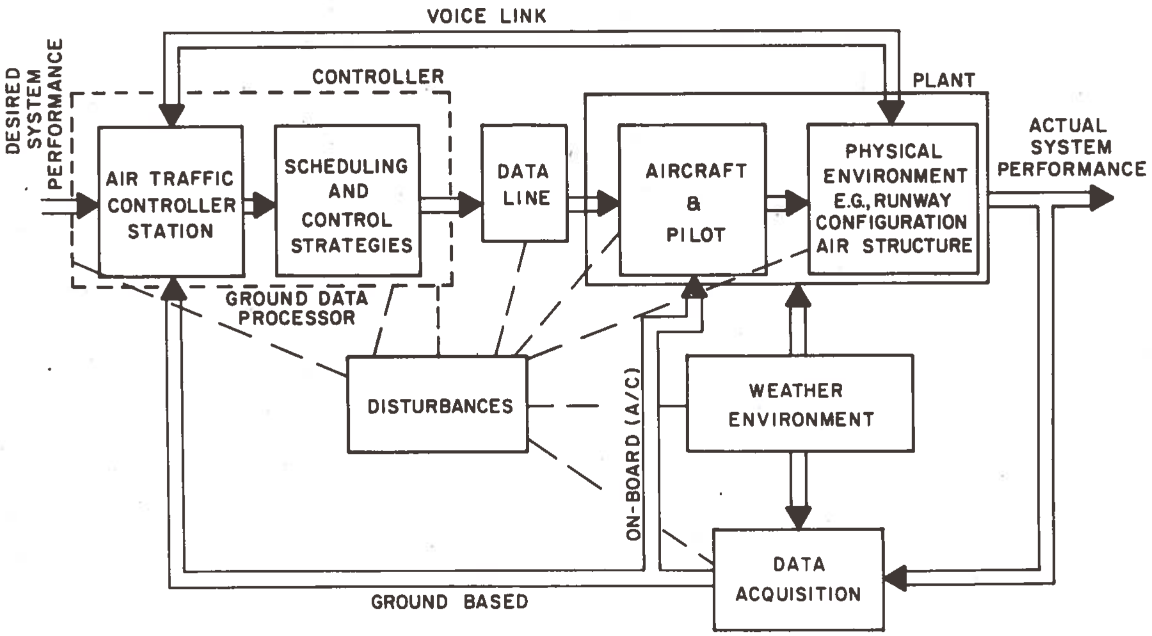

Figure 9. - ATC System Flow

APPLICATION TO MULTI-MODAL TRANSPORTATION SYSTEMS

It is envisioned that follow-on work will require a continuous augmentation of the Multi-Modal Automation Laboratory facilities and simulation software in order to simulate ground transportation systems as well as air transportation systems, e.g., dynamic bus routing simulations, personal rapid transit systems, and highway system simulation.

DYNAMIC BUS ROUTING SIMULATION

One of the promising transportation modes for urban transportation is Dial-A-Ride system (called, alternately, Dial-A-Bus, Demand-Actuated Route Transit, or Computer Aided Routing System) that consists of low-capacity vehicles operated over variable routes under variable schedules as determined by demand.[8] These systems have been conceived to meet the needs of low-density suburban transportation where travel distances are large and the demands are random. Computer simulation has been and will continue to be one of the major tools in investigating the interaction of requests, on-line computer, and vehicle fleet The development of control strategies for the collection and distribution of passengers may be aided by simulation.

The display system can be used to display vehicle position, projected schedule, vehicle breakdowns, and unsatisfied demands to the chief dispatcher. The proper distribution of tasks betwee the computer and dispatcher is an area that needs investigation. A properly structured facility can simulate the external disturbances, traffic densities, passenger demands, vehicle locatio dispatching programs, etc. A well designed and applied command and control system is essential for this type of transportation mode to be successful.

PERSONAL RAPID TRANSIT SYSTEMS

Personal rapid transit systems may be characterized as providing personal, demand-responsive, non-stop service over a relatively fine route web with small guided vehicles. Such systems may be used as city collection and distribution systems in conjunction with fast transit links to provide the commuter with a travel mode competitive with the automobile. The command and control problem areas related to this system are longitudina movement control, control at junctions, routing, dispatching, control of stations, and vehicle storage.

A method of developing and evaluating command and control concepts for these systems is by computer simulation. Before real alternative systems are implemented, large network simulations with many vehicles, realistic demand patterns, vehicle dynamics, and guideway interactions may be evaluated in terms of capacity, cost, and safety on this Multi-Modal Automation Laboratory.

HIGHWAY SYSTEM SIMULATION

Extensions and evaluations of the current Passing Aids Systems concepts can be facilitated by employing computer simulation. Several alternative methods are currently being developed to provide data to motorists on safe passing opportunities. Presentation of a simulated display of highway conditions during passing maneuvers to a driver of a vehicle may aid in assessing the trade-offs in effectiveness between alternate roadside and in-vehicle display techniques.

General highway engineering systems simulations provide economical preliminary solutions to the dynamic flow control problem. Typical applications are exemplified by expressway metering studies and the evaluation of traffic flow patterns in the face of impediments (car breakdowns, bus stopping and etc.). Also, the concept of minimum time versus minimum distance routing instructions for the motoring public can be analyzed employing appropriate simulation techniques.

BIBLIOGRAPHY

1. "Report of Department of Transportation Air Traffic Contro Advisory Committee", Volumes 1 and 2, December 1969.
2. Erwin, Ralph L., Jr., "Time-Sychronized Approach Control", Journal of the Institute of Navigation, Vol. 17, No. 3, 260 (1970).
3. Kuprajanow, A., Liebman, S., and Warshow, M., "Preliminary Evaluation of Selected Air Traffic Control Concepts", The Civil Aviation Research and Development Policy Study, November 1970.
4. Meyers, T., and Sutherland, W.R., "PDP-10 Air Traffic Simulator", Interim Report, Contract No. DOT-TSC-71, 1971.
5. Benham, F., Hatch, A., and Kass, S., "Investigation of ATC Man/Machine Techniques", Interim Report, Contract No. DOT-TSC-131, 1971.
6. National Aviation Facilities Experimental Center (NAFEC) Report, DS-FS-402, Rev. 2, "Digital Simulation Computer Program Functional Specification - Pilot Entry and Display Message Structure", June 1970.
7. "Programmers Reference Manual for the Dynamic Display Software System", Contract No. NAS 12-2051, 1970.
8. Wilson, N., "CARS - Computer Aided Routing Systems", School of Engineering, MIT Research Report #R67-12.

ACKNOWLEDGEMENTS

The authors wish to acknowledge the efforts of the Computer and Display Systems Branch personnel of the Transportation Syste Center, Department of Transportation. Also, the guidance and encouragement of Dr. F. Tung are appreciated.