A COMPUTER DRIVEN SIMULATION ENVIRONMENT FOR AIR TRAFFIC CONTROL STUDIES

E. A. Robin, R. S. Pardee, D. L. Scheffler, and F. C. Holland
TRW Computer Division
A. G. Halverson—Federal Aviation Agency

The technique of real-time simulation for studying air traffic control problems has been employed for many years. This technique has enabled investigators to simulate live air traffic situations for the purpose of evaluating new concepts and their relationship to the air traffic controller. The modern high-speed digital computer has increased the scope of these real-time simulation studies to include large scale air traffic control systems employing such computers in the control process.

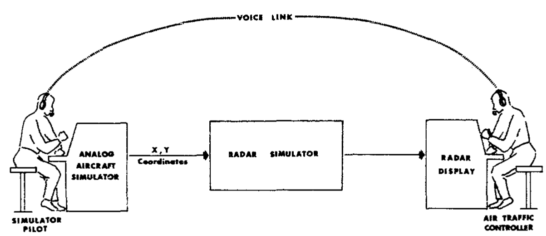

Figure 1. Analog Type Air Traffic Simulation.

Up to the present time each aircraft used in real-time simulation studies (conducted by the F.A.A. at the National Aviation Facilities Experimental Center) has been generated by a special purpose analog simulator. An individual "pilot" would be assigned to "fly" each simulated aircraft in a study. A radar simulator would transform these X, Y aircraft coordinates into proper signals to display this target on a standard radar display. (See Fig. 1.) The air traffic controller uses voice communication links to communicate with the simulator pilots.

Two of the major functions of a computer aided air traffic control system are scheduling and control. A sequence of aircraft is derived to deliver aircraft to a destination based on some criterion such as to safely maximize the aircraft acceptance rate at an airport, or reduce the average delay, controller workload, etc. The control function is then exercised to minimize the difference between the actual time the aircraft arrives at the destination and the schedule time. In the systems to be simulated this control is effected by the air traffic controller whose control decisions are based on computer-derived information displayed to him on some display media.

The increased controller workload caused by growth of air traffic and the influx of large high-speed jet aircraft in today's system necessitates quicker and more accurate decisions. Thus the computer is being proposed as an integral part of future air traffic control systems. Control suggestions may be derived by the computer and displayed to the controller for action. Many data processing functions, based on complex logic, must be performed in the process of scheduling and controlling aircraft. A general purpose digital computer with its flexibility can be easily used to implement this logic.

The question of "Why use a digital computer in real-time simulation?" may be answered in terms of its many advantages, e.g.:

Study of Computer Aided Systems. If the applicability of a digital computer in an air traffic control system is under investigation, then it follows that the simulation environment required to study solutions to this problem must also employ a digital computer. If more than one application is to be studied, then it is important that a general purpose simulation environment incorporating a general purpose digital computer should be available.

Generation of Aircraft Targets. Recent interest in introducing automation to terminal area near the airport air traffic control concepts has created stringent requirements on the realism of the analog aircraft simulators in terms of aircraft heading, position, and velocity. A digital computer can be used to generate these targets within this framework of requirements. It permits the inclusion of controlled navigation and speed error distributions so the effect of errors on the simulated system can be studied. The versatility of digital computer generated targets also allows additional functions to be programmed in the aircraft simulation that were unavailable in previous analog simulation equipment, e.g., the ability to realistically navigate the aircraft using the instrument landing system (ILS), or realistic simulation of take-off acceleration, etc.

Simulation of Sub-Systems. A digital computer can be used to simulate a large variety of complex sub-systems that might occur in present or future air traffic control systems. Simulation of these sub-systems avoids the necessity of developing these often complex equipment until their value has been reasonable well determined. Some examples of these sub-systems might be the radar or other form of position acquisition system, the radar or beacon tracking system, or a beacon attitude receiving and display system.

Flexible Programmable Situation Displays. Previous experience has shown that buffered general purpose displays are needed for controller and pilot situation displays. These units should also be capable of displaying tables and other alphanumeric information to the controller. By building the simulation around a general purpose digital computer the flexibility of a programmed display system with no "built in" format restriction can be readily obtained.

Data Collection and Analysis. The computer can be used to collect and analyze data while performing the other functions previously described. Thus the data collection can be integrated into any simulation.

Rapid Changes from Problem to Problem. A stored program enables rapid changes to be made from run to run. By changing the program an entire new simulation can be started with minimum effort in a few minutes. Thus one facility can be utilized on a time-shared basis to study many different air traffic control problems.

EQUIPMENT CONFIGURATION

Based on early experience it was decided that a large high-speed digital computer would be required to properly conduct these simulation experiments.[1] At that time the IBM 709 was installed at the National Aviation Facilities Experimental Center and was therefore recommended for use with this Computer Driven Simulation Environment (CDSE). Subsequently, the IBM 709 was replaced by an IBM 7090. The 7090 computer was equipped with two data channels,[2] a Direct Data Connection,[3] and a real-time digital clock.[4] (See Fig. 2.) The simultaneous read-write-compute feature of this computer made possible the transfer of large

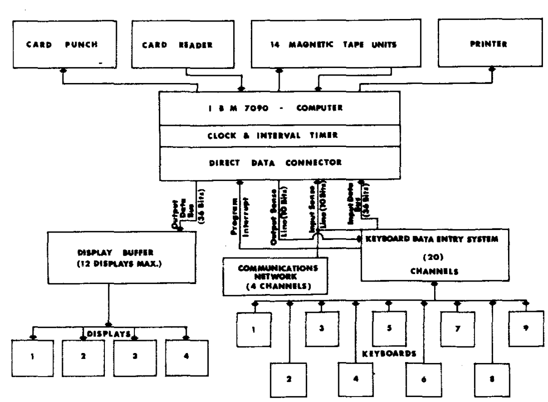

Figure 2. IBM 7090 Computer Equipment Configuration.

Figure 3. Charactron Display—Air Traffic Controller Position.

Figure 4. Simulator-Pilot T.V. Monitor Display.

blocks of data without excessively increasing program execution times. The Direct Data Connection provided high-speed data transfer to the Display Buffer system and from the Keyboard Data Entry system.

The displays chosen were of the Charactron type (19″ diameter) and had ability to display alphabetic and numeric information (fixed character size), special symbols (for aircrafts, fixes, etc.) as well as lines. Since the display format was under program control, flexibility existed for simulation on these displays for any desired situation (Fig. 3). Initially, there were four displays connected to the computer through the Display Buffer system. Three of these are presently used for air traffic controller consoles and one provides data for the simulator pilots by means of a closed circuit TV system. Four high resolution cameras transmit the data from the Charactron to six simulator pilots' TV monitor displays (Fig. 4).

A set of input keyboards are required to allow controllers and pilots to communicate with the computer. Six keyboards are used for simulator-pilot input functions (Fig. 5) and three for the air traffic controller positions (Fig. 6).

Four voice communication channels are provided between the controllers and the pilots. Each channel is monitored via the 7090 sense lines in order to derive the distribution of

Figure 5. Simulator-Pilot Input Keyboard.

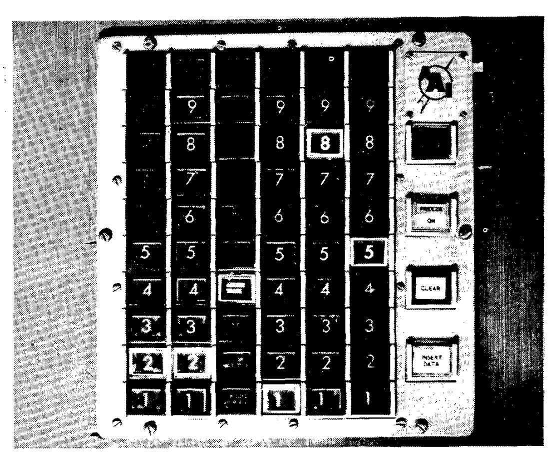

Figure 6. Air Traffic Controller Input Keyboard.

message lengths thereby enhancing the communication workload and delay studies.

PROGRAM ORGANIZATION AND TECHNIQUE

A major requirement in the selection of a programming language used was that it allow the programmer to utilize to the fullest extent the total capabilities of the IBM 7090 computer, including simultaneous input-output-compute, program interrupts, and all machine registers and instructions. At the same time it was desirable to have available a compiler which would facilitate the translation of mathematical statements into machine language since significant portions of simulation programs were to consist of mathematical formulae. Also, the use of a compiler would allow immediate programming contributions from highly technical personnel with limited programming experience.

The programming languages selected were the FAP Assembly Program and the FORTRAN Compiler. The main reason for the selection of these languages was that they were available and proven in daily operation. Programs written in either language could be used with those written in the other. Thus, persons of varying backgrounds and programming experience were able to work together.

An effort was made to write all real-time programs in the FAP assembly language and the off-line analysis programs in the FORTRAN language. The reason for this was that it was possible to write more efficient operational programs in machine language; and it was important to have real-time programs consume as little running time as possible, while this requirement was not necessary in the off-line programs.

For the purposes of these simulations the FORTRAN/FAP system proved to be adequate and imposed no limitations on the programmer's ability to exploit fully the capabilities of the computer. The problem of lengthy subroutine linkages which are inefficient in respect to both space in the computer memory and in running time during the operation of the problem was solved simply by using the COMMON pseudo-operation of FAP and FORTRAN. By the time a simulation program became operational, this list had grown to a size of over 10,000 words of memory, or about one-third of the total core memory. Since most subroutines made many references to data in the COMMON list, the use of this device saved thousands of memory locations. The maintenance of the COMMON list was at worst an inconvenience but certainly was not a hardship.

Three levels of interrupts were used in the simulation program. In order of decreasing priority they were the real-time clock interrupt, the external keyboard entry interrupt and the display transmission interrupt which was the 7090 data channel trap. The controllers or pilots entered information into the computer via the keyboards which triggered the keyboard entry interrupt. In general, no matter what function the computer was performing at the time, this function was interrupted for the time it took the keyboard interrupt program to read the keyboard message, check it for validity and store it in the message input buffer area of core; after which control was returned to the program in progress at the time of the interrupt. This whole process was completed in less than one millisecond. The clock interrupt operated in a similar manner and was set to interrupt the main program every ½ second. The third type of interrupt was used to transmit data to the displays without tying up the computer while sending this information. Had it been necessary to suspend other computer functions while updating displays, 20% of the available computing time would have been required.

One of the most significant subroutines in the Computer Driven Simulation Environment was the program to generate the simulated targets under the control of the pilot keyboards.[5] This technique resulted in a high degree of precision but more importantly, it permitted the inclusion of controlled navigation and speed error distributions. A number of additional functions were programmed in the digital aircraft simulation that were unavailable in previous simulation equipment.

Each aircraft flight plan was assigned a simulator number when it entered the problem.

A simulator assignment program maintained a dynamic list of available simulator numbers.

The simulator number was the key to all information concerning a given flight. Every flight had about fifty descriptive parameters associated with it, such basic things as X, Y, and Z coordinates, indicated air speed, performance characteristics, and many other parameters associated with the computer program. Each distinct parameter was stored in blocks of words equal to the number of simulators so that once the simulator was known, any piece of information concerning the flight could be obtained by indexing the initial location of the block by the simulator number.

For example, suppose a program required the altitude of the flight with the simulator number of 13. The desired altitude would be found in the 13th word of the block of altitudes associated with the simulators. This method of data storage proved to be very efficient in both target generation and display programs and provided a convenient method of reference between programs.

The collection of data on the movement of the simulated aircraft, performance measures and operational data was greatly simplified by using the same computer to generate targets and collect data. Data was written on magnetic tape during the simulation run and saved on tape for analysis after completion of the simulation. Tape buffering using the simultaneous write-compute of the 7090 Data Channel was used. This method made the control processing unit of the computer available at all times for the simulation program.

A TYPICAL SIMULATION PROBLEM

A typical simulation problem consisted of three main groups of programs arranged in a chain of three links. The purpose of Link 1 was to read into the computer some 94 parameters which specified the conditions of the run. On the basis of the parameters, input traffic flight plans were selected by the program from tape and stored away in the core. Also, data and tables needed by the operational control program were computed.

After completion of Link 1, the real-time simulation program, Link 2 was called. Link 2 consisted of 18,000 instructions and made use of a total of 28,000 memory locations. The initialization program (Link 1) ran for less than one minute compared to two hours or more for a simulation run (Link 2).

Link 2 consisted of the following main subroutines.

(1)	Generation of Simulated Aircraft	1900 instructions
(2)	Data Acquisition	160 instructions
(3)	Display	2000 instructions
(4)	Data Collection	1300 instructions
(5)	Keyboard Entry	1200 instructions
(6)	Operational Control Logic	3500 instructions

Upon completion of the simulation run the final data was recorded and Link 3 was called. Link 3 was read into the memory over Link 2. It consisted of some 52 different analysis programs to process the data collected during Link 2 and stored on magnetic tape.

Fig. 7 shows a functional diagram of the CDSE. The 7090 program consisted of four functional areas, simulated aircraft targets, data acquisition, operational control systems, and data collection. Some of the parameters read into the computer in Link 1 are shown in the figure. For example, the estimated aircraft characteristics, geometry of the area simulated, and the forecast wind are used in the operational control system logic. The characteristics of the data acquisition system, e.g., sampling

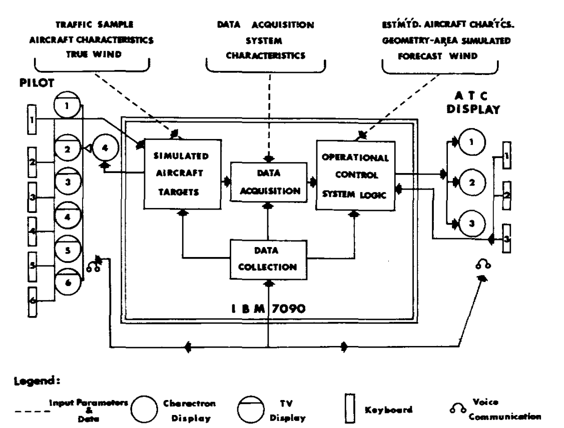

Figure 7. Functional Diagram of the Computer Driven Simulation Enviroment (CDSE).

rate of the radar or standard deviation of radar error, are used in the data acquisition subroutine. The external displays and keyboards provided the command and control hardware to integrate the air traffic controllers into the operational control system. They also allow the pilots to "fly" the simulated targets. Voice communication between the controllers and the pilots closes the control loop. The data collection program is able to monitor the communication lines, the simulated aircraft keyboard workload, and control system performance.

CONCLUSIONS

An extensive real-time simulation program can be handled most efficiently and expeditiously by maintaining a small group of specialists working closely together. Overall efficiency is increased if each individual in the group is given responsibility for a portion of the problem. This responsibility should begin with systems analysis and synthesis, and carry through to the flow charting, programming and debugging. The small group size and the full responsibility for major portions of the problem result in greatly reduced communication and coordinate requirements among members of the group. In addition, the knowledge and use of the computer machine language (FAP) or equivalent by all members of the group will make possible great savings in both program size and execution time.

The CDSE proved to be a versatile and reliable vehicle for carrying our real-time simulations of air traffic control systems. The keyboard and displays were both completely under program control, thus making it possible to change or improve formats in a matter of hours (or minutes) without any hardware modifications. Currently, two operational control systems are being evaluated. The ability to control displays and keyboards from the computer makes this use of the equipment possible.

The use of programmed digital aircraft simulators proved to be completely successful. With the digital simulators there was never any problems of drift or alignment that were present in the analog simulators.

The use of the computer for the set-up and operational control of the real-time simulation proved to be a reliable and efficient operation. Input traffic was generated by the computer and stored on magnetic tape. At the beginning of a run, the traffic sample number was entered in the computer console keys, and pre-punched cards containing the simulation parameters read into the card reader. These were the only human operations required since from this point on, the computer controlled the entire simulation. This method resulted in a smooth and efficient operation plus a considerable saving in set-up and analysis time.

BIBLIOGRAPHY

1. R. S. PARDEE, F. C. HOLLAND, E. A. ROBIN, et al., "Air Traffic Control Studies Report No. 10, Project TASC (Terminal Area Sequencing and Control), Final Report," TRW Computers Company, February 1961. Contract FAA/BRD-112) available from Office of Technical Services, U. S. Department of Commerce, Washington, 25, D. C.
2. IBM Reference Manual, "709–7090 Data Processing System".
3. IBM Special Systems Feature Bulletin, "Direct Data Connection for IBM 7090", RPQ M90976.
4. IBM Special Systems Feature Bulletin, "Core Storage Clock and Interval Timer", RPQ F89349.
5. F. C. HOLLAND, "A Digital Aircraft Simulation Program for the IBM 709–7090", TRW Computers Company ATC Technical Memorandum 62–10, September 1962.