REAL TIME SIMULATION OF ELEVATORS

George T. Hummet
Thomas D. Moser
Bruce A. Powell

1. INTRODUCTION

The principal transportation system in a highrise building is the elevator system. The performance of the system as perceived by the passengers can be measured by the time spent waiting for a car and smoothness of the ride. In turn, the waiting time is a very complex function of many variables: number of cars, car speed, number of floors served, passenger traffic, elevator dispatching strategy, etc. This paper discusses a recently developed minicomputer simulator, ELSIM, which simulates passenger traffic and the service rendered by a bank of elevators. This simulator uses a realistic mathematical model for passenger traffic within real-time minicomputer software and a display panel to produce a valuable tool for evaluating new elevator installations and modifications to existing systems.

The importance of a simulator to an elevator manufacturer lies in the desire to provide the optimum elevator configuration for each installation. This optimization process is made exceedingly complicated by the fact that passenger traffic varies from building to building. A multi-tenant office building, for example, will have a significant amount of in-and-out traffic, with very few people going from one upper floor to another (termed "interfloor traffic"). A single-occupant building on the other hand, will have extremely large traffic peaks at rush hours and will see a large amount of interfloor traffic. Even within a single building, traffic conditions vary continuously from morning up-peak to balanced off-peak to heavy two-way traffic at lunch time to evening down-peak. Considering these extreme cases along with such special building features as a cafeteria floor, convention floors and observatories, the elevator planner has found it necessary to treat each new or modified elevator installation individually. Elevators are expensive, and once a building is constructed, it is almost prohibitively expensive to add more cars. The ELSIM simulator is a tool that can be used to optimize the initial system configuration.

This paper indicates how the ELSIM simulator is applied to evaluate a proposed elevator system. A detailed description of the mathematical model for passenger generation and a general description of the minicomputer software are also given.

2. OVERVIEW OF THE SIMULATOR

The responsibility of providing good elevator service to an arriving stream of passengers is that of a supervisory dispatching strategy. Its logic monitors all incoming calls, determines the status and locations of the cars and then assigns cars to calls. Since passenger demand contains a degree of random variability, realistically estimating passenger waiting time requires techniques other than table look-up and analytical formulas for average conditions. Computer simulation provides a convenient way to study the complex interaction of passenger traffic and the dispatching strategy.

A key portion of the ELSIM system is the interactive display control panel, which shows the status of all elevators, car calls and unanswered corridor calls. Using logic from a computerized dispatcher and driven by a minicomputer, the panel displays corridor and car calls as well as status for each elevator. The panel may be operated either locally in conjunction with the dispatcher/minicomputer portion of the system or remotely via direct-dial phone line. This feature permits elevator consultants and building architects to use the panel for evaluating the performance of proposed elevator systems from their own offices with only a phone line connection to the simulation logic.

The ELSIM simulator is used in three distinct ways: (i) estimating performance for new or modified elevator installations, (ii) improving dispatcher strategy and (iii) verification of production software. In the process of evaluating proposed elevator installations, the user will specify all system parameters (number and speed of cars, number of floors, general traffic conditions, etc.) and make a simulation run. A simulated stream of passengers is generated, and the resultant corridor-call lights are indicated on the panel. The status and motion of the cars in response to the passenger demands can be seen on the panel's column of lights. At the end of the run, waiting time statistics are printed on the terminal device. Usually, several distinct traffic periods (e.g., morning, noon, evening rush hour) are run, and the results evaluated. Also, sensitivity studies are done in which the effect of, say, one car out of service or a sudden surge in traffic intensity could be estimated.

Over the past 25 years, elevator control has gone from cars controlled manually by attendants and dispatched from the lobby by a "dispatcher" to electronic car controllers and sophisticated electronic dispatchers. The most recent systems have solid-state car controllers and computerized supervisory dispatching controllers. The process of improving the dispatching strategy is not straightforward. It is done by an experienced elevator engineer who has an intimate knowledge of the strategy and can recognize when improvements can be made by observing the lights on the panel.

The ELSIM simulator goes a step beyond the traditional event-based digital simulation program. In general, we are faced with the task of verifying the accuracy of simulated control algorithms by displaying the simulated responses in a manner such that those familiar with the actual system may quickly and surely interpret the results. Traditional digital simulation uses a special purpose simulation language and provides a printout of the performance of the simulated system. The section of code that contains the dispatcher strategy is usually complex and difficult to change. With ELSIM, the user can observe the real-time dynamic motion of the elevators in response to both automatically generated passenger traffic and manually inserted calls, and the summary of waiting-time statistics are printed out. Also, the elevator dispatcher used in the simulator is a production line version that could be used for real-time control in an actual building. Because of these features, the ELSIM system has gained acceptance by the ultimate user.

As shown in Figure 1, the system consists of several subsystems: (a) minicomputer, (b) computerized elevator dispatcher, (c) display panel, (d) keyboard/printer and (e) magnetic tape unit. A photograph of the entire system is shown in Figure 2, which appears at the end of this paper.

A general description of the elevator simulator includes the following list of functional objectives:

(a) accept commands issued by the dispatcher
(b) generate responses to those commands
(c) display the results on an interactive I/O panel
(d) accept corridor and car calls entered manually on the panel
(e) automatically generate realistic patterns of corridor and car calls corresponding to the demand period of interest
(f) provide a measure of the system's performance by compiling waiting time statistics
(g) enable the user to specify the pertinent parameters used to simulate car dynamics such as car speed, door times, floor height, etc.
(h) permit the display panel to be operated either locally via hard-wired connections or remotely via standard phone line/modem connections.

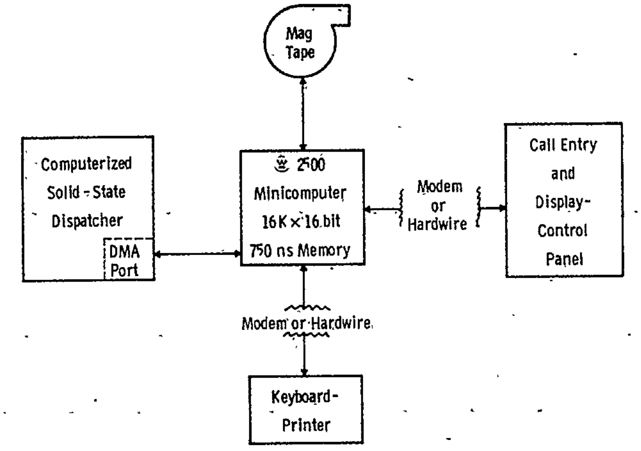


Figure 1. General Block Diagram of ELSIM system

3. SAMPLE SIMULATION RUN

The input to the simulator is accomplished in a conversational mode on the terminal. Input quantities include passenger traffic information, building parameters and elevator system parameters. Traffic data includes (a) per floor population, (b) relative use of the elevators by each floor, (c) traffic intensity, (d) percent of passengers with main-floor origin, (e) percent of passengers with main-floor destination and (f) percent of inter-floor traffic. Elevator system parameters include number of cars, capacity of each car, car speed and acceleration, door operation time and various aspects of the dispatching strategy. Building parameters include floor-to-floor height and number of floors served. Figure 3 shows a listing of a set of input parameters for a hypothetical 14 story building.

After the input has been entered, the user can make a simulation by pressing a "RUN" control button on the panel. As the simulated passengers register corridor calls, the panel's call lamps will light. As cars respond to the calls, the position of the cars will be shown, and the call lamp will be turned off as the call is cancelled. The pattern of calls and car responses is shown on the panel, and an experienced user can discern features of the dispatcher strategy. Finally, at the end of a simulation period (usually 15 minutes), a print-out of the waiting times is made on the terminal. This print-out includes average and maximum waiting time for two classes of calls:

(i) up-calls from the main lobby and (ii) all other calls. Also, the distribution of calls in 5 and 10 second intervals is tabulated. Figure 4 shows a sample statistics print-out corresponding to the simulation run specified in Figure 3.

4. DESCRIPTION OF THE SOFTWARE

Panel

In order to strengthen the analogy between elevator cars and illuminated indicators, information is arranged in an eight column by thirty-two row pattern wherein the eight vertical columns represent eight cars and the thirty-two rows represent the corresponding floor positions. As shown in the detailed layout of Figure 5, corridor call buttons are arranged in two (up calls, down calls) vertical columns immediately alongside the car-columns. Two illuminated indicators and one illuminated pushbutton switch are located at each floor position for each car. They represent location (amber light), dispatcher assignment (green) and car-call (white), respectively. Below each vertical column an additional array of white, amber and red indicators/switches is reserved for displaying additional per car status information such as hall lanterns, load and various indications of the operating state (In Service, Doors Open, Up Service, etc.) The remaining panel buttons are arranged in several functional groups of Control, System Timers and Special Corridor-Call Buttons.

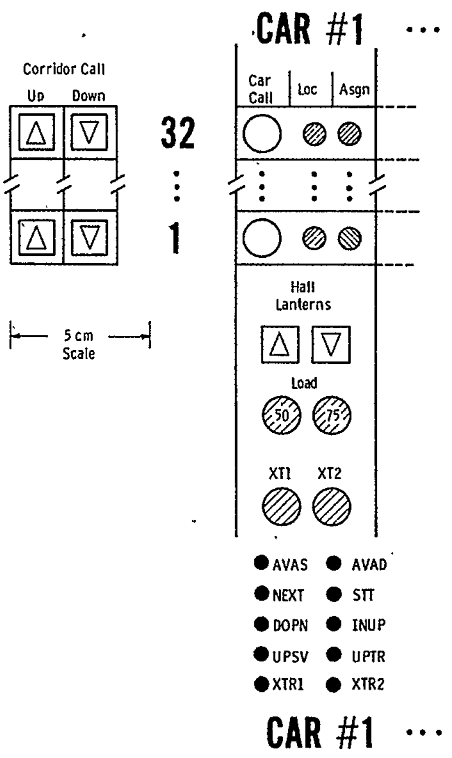
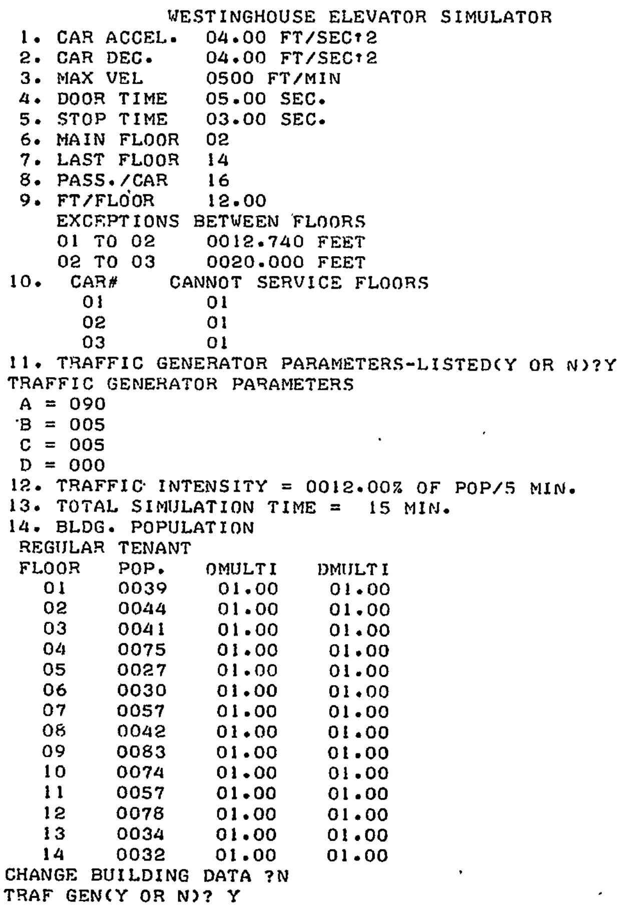
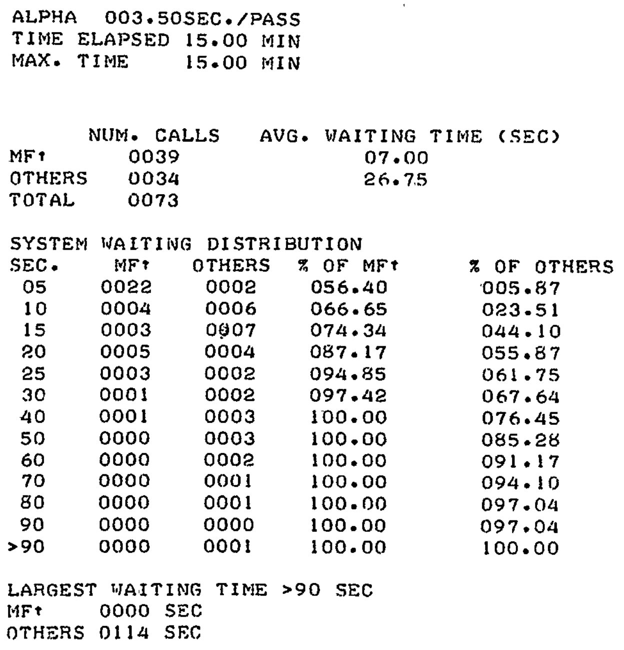


Figure 5. Layout of ELSIM panel



```
                WESTINGHOUSE ELEVATOR SIMULATOR
 1. CAR ACCEL.  04.00 FT/SEC↑2
 2. CAR DEC.    04.00 FT/SEC↑2
 3. MAX VEL     0500 FT/MIN
 4. DOOR TIME   05.00 SEC.
 5. STOP TIME   03.00 SEC.
 6. MAIN FLOOR  02
 7. LAST FLOOR  14
 8. PASS./CAR   16
 9. FT/FLOOR    12.00
    EXCEPTIONS BETWEEN FLOORS
    01 TO 02    0012.740 FEET
    02 TO 03    0020.000 FEET
10.  CAR#     CANNOT SERVICE FLOORS
      01        01
      02        01
      03        01
11. TRAFFIC GENERATOR PARAMETERS-LISTED(Y OR N)?Y
TRAFFIC GENERATOR PARAMETERS
 A = 090
 B = 005
 C = 005
 D = 000
12. TRAFFIC INTENSITY = 0012.00% OF POP/5 MIN.
13. TOTAL SIMULATION TIME =  15 MIN.
14. BLDG. POPULATION
 REGULAR TENANT
 FLOOR   POP.    OMULTI   DMULTI
   01    0039     01.00    01.00
   02    0044     01.00    01.00
   03    0041     01.00    01.00
   04    0075     01.00    01.00
   05    0027     01.00    01.00
   06    0030     01.00    01.00
   07    0057     01.00    01.00
   08    0042     01.00    01.00
   09    0083     01.00    01.00
   10    0074     01.00    01.00
   11    0057     01.00    01.00
   12    0078     01.00    01.00
   13    0034     01.00    01.00
   14    0032     01.00    01.00
CHANGE BUILDING DATA ?N
TRAF GEN(Y OR N)? Y
```

Figure 3. Typical set of input parameters for elevator simulation



```
ALPHA   003.50SEC./PASS
TIME ELAPSED 15.00 MIN
MAX. TIME     15.00 MIN


       NUM. CALLS    AVG. WAITING TIME (SEC)
MF↑       0039                07.00
OTHERS    0034                26.75
TOTAL     0073

SYSTEM WAITING DISTRIBUTION
SEC.     MF↑     OTHERS   % OF MF↑      % OF OTHERS
 05     0022     0002     056.40         005.87
 10     0004     0006     066.65         023.51
 15     0003     0007     074.34         044.10
 20     0005     0004     087.17         055.87
 25     0003     0002     094.85         061.75
 30     0001     0002     097.42         067.64
 40     0001     0003     100.00         076.45
 50     0000     0003     100.00         085.28
 60     0000     0002     100.00         091.17
 70     0000     0001     100.00         094.10
 80     0000     0001     100.00         097.04
 90     0000     0000     100.00         097.04
>90     0000     0001     100.00         100.00

LARGEST WAITING TIME >90 SEC
MF↑    0000 SEC
OTHERS 0114 SEC
```

Figure 4. Simulation output: Statistical distribution of waiting times

Information transfer between the panel and minicomputer is based upon a simple, yet reliable scheme where depressing a panel push-button (momentary switch action) causes the panel to transmit a unique push-button address to the computer, and the computer responds by transmitting a command to the panel that turns on the appropriate push-button lamp. Since this entire sequence requires less than several hundred milliseconds, the push-button is usually illuminated before the user can retract his finger from the button. Also since lamp control commands are stored in a flip-flop array within the panel, an illuminated button can be turned off only by an appropriate computer command. In addition to illuminated buttons, the computer also controls the various panel indicator lamps in exactly the same manner: commands sent by the computer are decoded and steered to indicator flip-flops that retain each lamp's state until another command changes it.

An asynchronous, full-duplex communications channel permits rapid bi-directional information exchange between the panel and minicomputer either locally via a twisted pair connection at 6.3 KBaud or remotely via phone lines and modems at 300 Baud.

Dispatcher

The dispatcher used in the ELSIM system is an actual production-line version of a special purpose minicomputer designed to interface with a solid-state elevator control system. A programmable core memory with a paper tape interface facilitates rapid re-programming to alter the dispatcher logic. The dispatcher also contains a real-time clock used for timing the frequency of occurrence of certain events and establishing control intervals during which priorities of calls are altered from the normal. The dispatcher, for example, can respond to up-traveling, fully loaded cars leaving the main floor by establishing a "Heavy Up-Peak" period for, say, three minutes during which main floor calls receive a higher than normal demand priority within the dispatcher's assignment algorithms.

A direct memory access (DMA) port to the dispatcher core memory is the main communications link between the dispatcher and the outside world. In an actual elevator installation the DMA port is connected via bi-directional data links to each car controller and corridor call entry system. The simulated system also transfers information through the same DMA port. The minicomputer periodically transmits vital per-car and system information directly into and extracts car-controller commands from the dispatcher DMA port. These commands determine each car's simulated motion and response to demands created by corridor and car calls. The dispatcher, then, receives information and issues commands in a manner analogous to actual on-line elevator control, while the simulator computer generates realistic responses to these commands.

Since the minicomputer does not perform any assignment computations normally performed within the dispatcher, errors due to approximated dispatcher algorithms are entirely eliminated in the ELSIM simulator. This greatly facilitates and expedites checkout of standard dispatcher programs as well as development of new dispatching features.

Minicomputer and Software

The minicomputer portion of the ELSIM system consists of a Westinghouse 2500 with 16K x 16 bit, 750 ns memory, a real-time clock (RTC), and three peripherals: a high speed paper tape reader (HSPTR); a 9 track magnetic tape drive and a keyboard printer. One general purpose interface card containing a single 16 bit parallel I/O port is used for both dispatcher and panel communications. The HSPTR is required for reading assembly-language programs prepared on a W-2500 cross assembler. Loaded via a resident linking loader and stored on magnetic tape, the assembly-language programs that comprise the ELSIM system are overlaid as needed by a small resident Overlay Executive.

A general block diagram in Figure 6 shows the main software structure of the ELSIM system. As illustrated in the figure, the entire system is interrupt-driven from an IDLE state to perform tasks in response to either panel or RTC interrupts. This structure permits the user to interrupt a simulation run in order to restart, respecify or print statistics merely by pushing the appropriate panel control button. A summary of panel control functions is given in Table I.

A typical sequence of operating steps begins with the Interactive program when the user, after pushing the "C1" control button, specifies the simulation variables by typing values for each of the parameters requested by the program. A representative set of values is given in Figure 3.

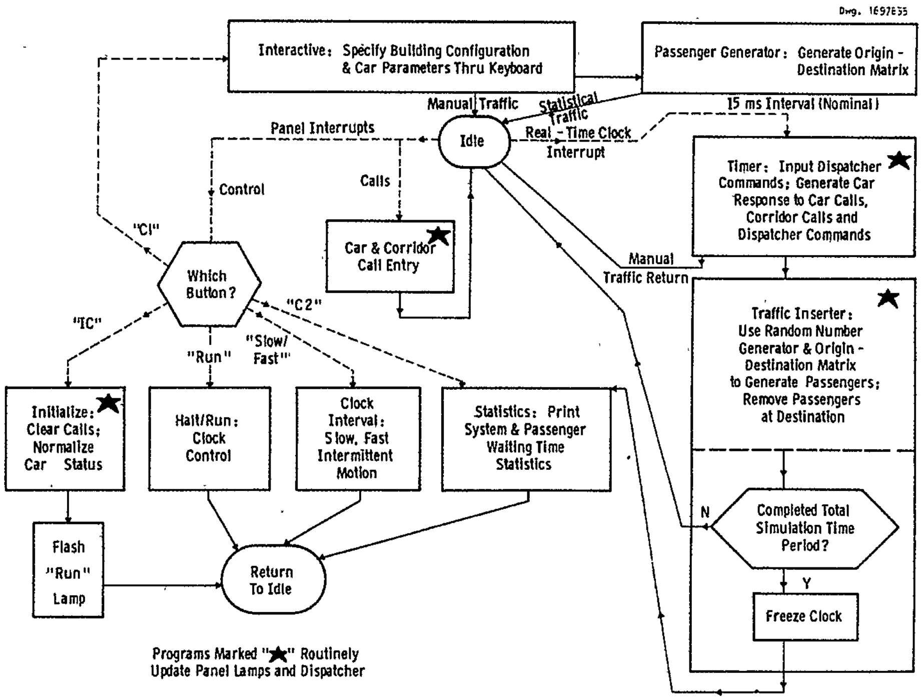
The user's response to the last query in the list, "Traffic Generator", determines whether or not the program exits to the passenger generator program prior to returning to IDLE. When called, the passenger generator utilizes several interactive input parameters to construct and store two probability matrices that serve as the basis for dynamically generating passengers later during the actual simulation time period.

Exit from the interactive program is first to the traffic generator and then (via clock interrupts) to the timer, inserter, and statistics-print programs. The final step in the pre-simulation sequence is the execution of an initialization subroutine that clears call and statistics tables, and resets all cars to a normalized state of not moving, doors closed, etc. A flag set by this subroutine signals the IDLE routine (see Figure 6) to flash the RUN lamp at a rate of about 30 per minute.

TABLE I

SUMMARY OF PANEL CONTROL FUNCTIONS

BUTTON		Action
ON	...	Power ON/OFF to panel
RUN	...	Halt/Run .. enables and disables clock interrupts, thereby determining either "RUN" or "HALT" state of simulation
FAST	...	Fast Motion .. increases simulation speed by decreasing time intervals between interrupts from nominal 15 ms to 7.5 ms
		Intermittent .. if both "FAST" and "SLOW" are depressed, then the simulation proceeds the normal speed for several seconds, then halts. This cycle repeats when the "RUN" button is depressed
SLOW	...	Slow Motion .. decreases simulation speed by increasing interrupt intervals to 30 ms
IC	...	Initial Condition .. removes all calls and restores cars to quiesent state of not moving, doors closed and no lantern
C1	...	Reinitialize .. calls the interactive program, which permits altering either building and/or simulation parameters
C2	...	Print Statistics .. halts the simulation and prints statistics based upon current values of average waiting times and number of corridor calls. Simulation resumes when the "RUN" button is depressed
C3 (SPARE)	...	



Programs Marked "★" Routinely Update Panel Lamps and Dispatcher

Figure 6. ELSIM Software structure - Dashed lines indicate interrupt-driven program entries

This signals the user that initialization is complete and the system is ready to proceed with the simulation interval. When the RUN button is depressed the system proceeds from HALT to RUN by enabling RTC interrupts.

Actual simulation of car motion and responses occur, then, only while the system is in the RUN state. Driven by 15 ms interrupts, the timer program steps through the eight system cars such that each car's status and position are updated every 8 x 15 = 120 ms. This timing sequence is followed even when fewer than eight cars are in the simulated elevator bank: the user removes unwanted cars simply by pushing the Out-of-Service buttons located in the per car column on the panel, while the timer merely exits back to idle when it encounters an Out-of-Service car. The 120 ms timing resolution, therefore, remains in effect for all per-car calculations performed within the timer program.

For each In-Service car the timer checks for both car and corridor calls ahead of and behind the car's position, where "ahead" and "behind" are defined with respect to the car's travel direction. The timer program utilizes dispatcher commands when computing a car's response. Thus if a car is given an assignment by the dispatcher to answer a particular corridor call, such as UP call at floor #10, the timer first checks the car's current status (In-Service, Moving, Doors Closed. ...) and then computes a response for that car. While a car is moving, its position is calculated using:

$$\Delta s = \frac{1}{2}a\Delta t^2 \quad \text{.... during acceleration interval}$$

$$\Delta s = V_0\Delta t \quad \text{.... uniform motion interval}$$

$$\Delta s = -\frac{1}{2}d\Delta t^2 \quad \text{.... during deceleration interval}$$

where

Δs = incremental position change (feet)

a, d = acceleration, deceleration rates

V_0 = maximum velocity

Δt = time increment (120 ms)

This calculation yields s, the car's absolute location in the building. Using "s" and a table containing absolute floor locations, the timer calculates the next floor at which the car could stop if it were to decelerate at the normal rate, d. This floor, which in elevator vernacular is termed the "Advanced Carriage Position" or "ACP", is the floor position reported by the timer to panel's position indicators. Thus car motion as viewed on the panel is exactly analogous to car position indicators in an actual elevator system, where electronically derived ACP signals are used to determine the next floor at which the car could stop if it receives a dispatcher assignment or car call while moving.

On the simulator, then, we observe a car's motion via illuminated position indicators showing the ACP as it advances from floor to floor. If we suddenly insert a car call at the ACP floor the car responds - just as in an actual system - by slowing down, stopping and opening the doors at that floor. Of course, if the call is placed well ahead of the car's position, it will not slow down until the ACP advances to that floor.

If, as is usually the case for simulated traffic studies, the user has requested automatic traffic generation, the inserter program runs as an adjunct to the timer at the 15ms (real time clock) interrupt intervals. Using a random number generator in conjunction with the origin-destination matrices previously created by the Traffic Generator program, the inserter finds the next passenger's entry time. Then in a subsequent pass through the program in which simulated real time equals that passenger's entry time, the inserter once again uses the random number generator along with the origin-destination matrices to find the passenger's entry and destination floors. This technique of using statistically derived probability matrices for generating passengers in real time eliminates the need for reserving large portions of memory to store a complete passenger list, as would be required if the list were generated prior to the simulation run.

Waiting-time intervals are tabulated and stored by the inserter. Later, either at the completion of the specified traffic study time period or in response to the "C2" panel button (see Figure 6), these data are used by the statistics-print program to derive and print the appropriate statistics.

In order to complete the discussion of the panel control buttons, we refer the reader once again to Figure 6 and also to a summary of Control Button Functions listed in Table 1 where we see that the user may at any time interrupt the normal IDLER*TIMER*INSERTER program execution sequence to perform the following tasks: Reinitialize ("IC" button); Halt ("RUN" button); Slow/Fast Motion (two buttons, "SLOW", "FAST"); or Intermittent Motion (depressing both "SLOW" and "FAST"). This last state, where the clock runs for about five seconds, then freezes until the "RUN" button is depressed, then runs for five seconds, etc., is particularly useful for debugging new dispatcher features, where the user is interested in carefully analyzing the sequence of dispatcher assignments as the cars change status. "Slow" motion is also useful for this purpose.

5. PASSENGER TRAFFIC GENERATOR

The mathematical model that underlies the simulation of passenger traffic attempts to reflect the variability that exists in real-life traffic streams. The variability is seen in two general areas:

1. The traffic intensity as measured by the number of passengers using the elevators in a 5-minute time period varies throughout the day.

2. The distribution of calls with respect to origins and destinations will vary from morning to evening.

The passenger model will be presented so that appropriate parameters can be entered and the program will generate the proper traffic pattern.

If we were an omniscient observer of an actual elevator system in operation, we would see a stream of passengers pushing buttons. A hypothetical log book might show the following information:

Identification of Passenger	Time of* Call Registration	Floor of Origin	Floor of Destination
# 1	7:45:03 am	1	8
# 2	7:45:15	1	5
# 3	7:45:31	3	1
⋮	⋮	⋮	⋮
# n	12:11:08 pm	12	7
#(n+1)	12:12:12	1	3
⋮	⋮	⋮	⋮

---

* Time is in HR:Min:Sec.

It is precisely this information that the passenger generator provides to the ELSIM simulator. We can describe the random processes underlying the arrivals of passengers to the elevator system using simple probability concepts.

First, consider the number of passengers using the elevators in a given period of time. The Poisson process is a well-known stochastic process which has been well suited for counting the number of occurrences of an event when the occurrences are generated randomly. It has been shown that the generation of such diversified things as radioactive particles, telephone calls, failures in electronic equipment and automobile traffic can be satisfactorily modeled with the Poisson process. The fact that the same technique could be used to model newly arriving passengers to an elevator system was shown by the authors in actual observation in dozens of buildings. The Poisson probability law is easily stated:

$$\text{Probability}\left\{\begin{array}{l}\text{n passengers are}\\ \text{generated in time T}\end{array}\right\} = \frac{(\lambda T)^n e^{-\lambda T}}{n!}$$

where λ = arrival rate (average number of passengers per unit time) and T is a given period of time. It follows that the distribution of time between passenger arrivals is negative exponential, with mean time between arrivals of $1/\lambda$ time units. The ELSIM simulator implements the generation of the times of arrivals of successive passengers 1, 2, 3, ... by the following recursive formula:

$$t_o = 0$$

$$t_i = t_{i-1} + \left(\frac{-\ell n(r)}{\lambda}\right) \quad i = 1, 2, 3, \ldots$$

where t_i = time of arrival for the i-th passenger and r is a random number on the interval (0, 1). Since traffic intensity is typically expressed as a percentage of the population in a 5-minute period, the value of λ is given by

$$\lambda = \frac{0.01TI}{300} \sum_i POP_i$$

where TI is the given traffic intensity, POP_i is the population of floor i and the units of λ is in passengers per second.

The above formulas have not considered the pattern of origins and destinations. We use information on the expected use of the building to construct an <u>origin density vector</u> and an <u>origin-destination matrix.</u> The i-th element of the origin density vector measures the percentage of new passengers whose floor of origin is floor i. Of all passengers with origin at floor i, the i-j^{th} element of the origin-destination matrix measures the percentage going to floor j. Consider the following example for a 4-story building, where Figure 7-A shows the origin density vector and Figure 7-B shows an origin destination matrix.

	i=1	2	3	4
FROM FLOOR i:	40%	24	12	24

Figure 7-A. Origin Density Vector

	To FLOOR			
	j=1	2	3	4
FROM FLOOR i = 1	0	40	20	40
2	67	0	11	22
3	67	16	0	17
4	67	22	11	0

Figure 7-B. Origin-Destination Matrix

Once these two items have been calculated, it becomes a simple matter to determine floors of origin and destination for each passenger with Monte Carlo sampling.

To describe how these traffic parameters are constructed in the general case, we must first recognize that traffic to and from a floor is proportional to its population. Second, we recognize that a passenger may be one of 3 types:

(a) up travelling passenger with origin at the main floor

(b) down travelling passenger with destination of the main floor

(c) interfloor passenger with both origin and destination above the main floor.

Assume the following notation:

A = % of all passengers that are type (a)

B = % of all passengers that are type (b)

C = % of all passengers that are type (c)

ORIGIN(i) = % of all passengers with origin at floor i

OD(i,j) = % of passengers going to floor j, given origin is floor i

POP(i) = population of floor i

ORIGIN(·) is the origin density vector, and OD(·,·) is the origin-destination matrix. The following formulas are used to compute ORIGIN(·) and OD(·,·) as a function of A, B, C and POP(i). Without loss of generality, the main floor is called floor 1 and the other floors are labelled 2, 3, ..., N. First, compute

$$\rho_i = \mathrm{POP}(i) / \sum_{i=2}^{N} \mathrm{POP}(i)$$

Then, the origin density vector is completely specified by

$$\begin{cases} \mathrm{ORIGIN}(1) = A & \\ \mathrm{ORIGIN}(i) = (B+C)\rho_i & i = 2, 3, \ldots, N \end{cases}$$

The origin-destination matrix is somewhat more complicated:

$$\mathrm{OD}(1,j) = \begin{cases} 0 & \text{if } j=1 \\ 100\,\rho_j & \text{otherwise} \end{cases}$$

$$\mathrm{OD}(i,1) = \begin{cases} 0 & \text{if } i=1 \\ \frac{B}{B+C}\,100 & \text{otherwise} \end{cases}$$

$$\mathrm{OD}(i,j) = \begin{cases} 0 & \text{if } i=j \\ \frac{C}{B+C}\,100P_{ij} & \text{otherwise} \end{cases}$$

where

$$P_{ij} = \mathrm{POP}(j) / \sum_{\substack{k=2 \\ k \neq i}}^{N} \mathrm{POP}(k) \quad .$$

As an example of the above formulas, consider a balanced traffic pattern with

A = 40
B = 40
C = 20

and a 4-story building with floor populations of 50, 200, 100 and 200. The values for ρ_i are $\rho_2 = 0.4$, $\rho_3 = 0.2$, $\rho_4 = 0.4$ and

$$\begin{aligned} \mathrm{ORIGIN}(1) &= A &&= 40 \\ \mathrm{ORIGIN}(2) &= (B+C)\rho_2 &&= 24 \\ \mathrm{ORIGIN}(3) &= (B+C)\rho_3 &&= 12 \\ \mathrm{ORIGIN}(4) &= (B+C)\rho_4 &&= 24 \end{aligned}$$

Note that these numbers are shown in Figure 7-A. The reader can verify that the formulas for OD(·,·) will give the results of Figure 7-B.

It should be noted that in actual practice, the traffic is not always proportional to actual floor populations. For example, a sparsely populated computer center floor would typically generate much more traffic than a floor of 200 order-entry clerks. In this case, multipliers are used so that traffic is proportional to an "effective" population.

It is obvious that a single set of values for ORIGIN(·) and OD(·,·) are good for only one distinct traffic pattern. Other patterns are developed by altering the basic parameters A, B and C. The following table lists some typical traffic patterns and parameter values.

Pattern	A	B	C
Morning up-peak	90	5	5
Mid-morning off peak	45	45	10
Noon- go to lunch	20	60	20
Noon-return from lunch	70	10	20
Evening down peak	5	90	5

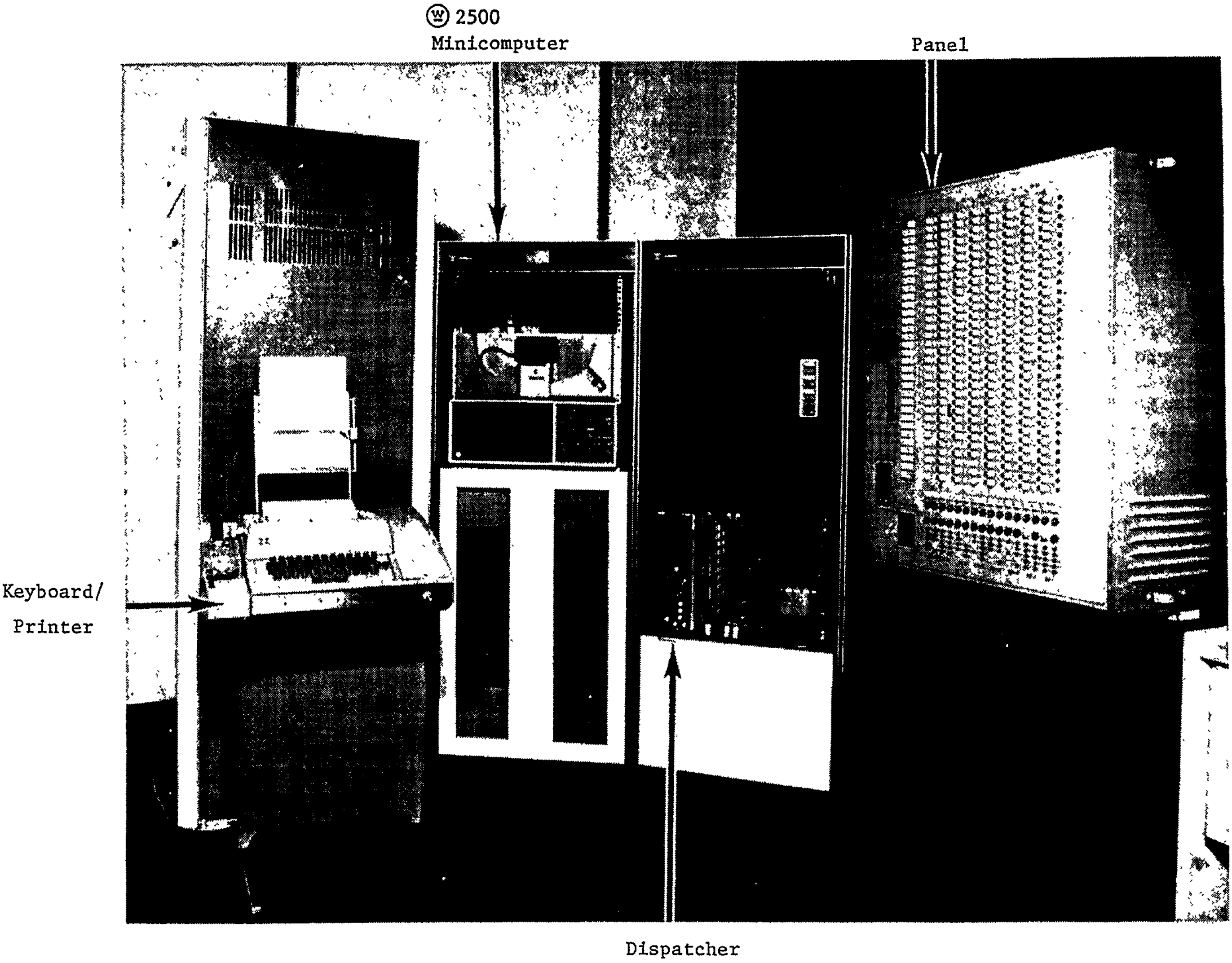


Figure 2. Photograph of the ELSIM system

CONCLUSIONS

The development of the ELSIM system represents the use of a realistic mathematical model for passenger traffic, minicomputer software and display panel to produce an important engineering tool. The system has been used to plan many new elevator installations and to develop improved dispatching strategies. Real-time simulation and visual display of pertinent system indicators has been widely accepted by elevator engineers, building owners, architects and consultants. The real-time ELSIM system has provided an invaluable extension to traditional event based digital simulation.

ACKNOWLEDGMENT

During the development of elevator simulation at Westinghouse R&D Center, a number of individuals have made significant contributions. They are R. A. Altekruse, D. M. Edison, A. F. Kirsch, M. Kurland, A. Popp, H. C. Savino (dec.), D. H. Shaffer and D. P. Wei.

REFERENCES

1. Browne, J. J., and J. J. Kelly, "Simulation of Elevator System for World's Tallest Buildings", Transportation Science, 2, 35-56 (1968).

2. Powell, Bruce A., "Elevator Banking for High Rise Buildings", Transportation Science, 9, No. 3, (Aug. 1975).

3. Powell, Bruce A., and D. P. Gaver, "Variability in Round-Trip Times for an Elevator Car During Up-Peak", Transportation Science, 5, No. 2, 169-179 (May 1971).

4. Strakosch, G. R., Vertical Transportation: Elevators and Escalators, Wiley, 1967.