MAINTAINABILITY ENGINEERING SIMULATION SYSTEM: AN APPLICATION OF DISCRETE SIMULATION TO ENGINEERING EDUCATION

Ronald S. Morris
USAMC Intern Training Center - USALMC
Red River Army Depot
Texarkana, Texas

Section I

Introduction

In the instruction of system engineering disciplines, like maintainability or reliability, the importance of considering the interactions of all system parameters in the evaluation of proposed hardware configurations must be emphasized. The student should thoroughly comprehend that the nature of these interactions determines the impact of a design configuration on performance.

For example, he must realize that additional money spent on improving reliability and maintainability would have to be evaluated against the expected increase in system performance and decrease in operating cost. The performance, however, is also dependent on the number of maintenance personnel available for servicing and corrective maintenance and other resources required to operate the system hardware. Because all these parameters may be at optimal values, the additional money spent on reliability/maintainability will provide little or no improvement in performance and may, in fact, increase operating cost.

To promote an understanding of these complex interactions a teaching aid in the form of a simulation model can be employed. The paper describes a maintenance/reliability simulation model that has been used to introduce the concepts of maintainability/reliability engineering to undergraduate engineering students and that will be used in the training of graduate level systems engineering students. The graduate level personnel will have studied the analytical processes involved in systems engineering and will have a considerable background in statistics prior to use of the simulation. For this level student, use of the model also provides an excellent introduction to the techniques of digital simulation. The undergraduate student, on the other hand, cannot use sophisticated analytical or statistical techniques to suboptimize parameters and must rely on intuition and experience in establishing input parameters values. In either case, the use of the model provides the student an excellent understanding of the complex interactions between various parameters of the system. The model will be described in Section II. Section III includes a sample and description of the output and a list of required input variables. Section IV contains a brief discussion of the use of the model as a teaching aid.

Section II

Maintainability Engineering Simulation System

The Maintainability Engineering Simulation System (MESS) is a digital simulation model of an emergency-type service using expensive or complex hardware systems. Examples of appropriate service operations which could be simulated are an ambulance service, a medical helicopter operation, an air/sea rescue service, a fire department, an emergency utilities repair, etc. Whatever the service operation chosen, it must generally conform to the following operational concepts.

Operationally-ready vehicles reside in a pool or holding area until such time as a call arrives and the vehicle is dispatched. As a consequence of the dispatch function, the vehicle is removed from this ready pool. In the event a call arrives, but no vehicles are available in the ready pool, the call will be placed in a queue of waiting calls. This queue is generally a first-in-first-out ranked queue. As soon as a vehicle arrives in the ready pool the awaiting-calls-queue is queried: in the event there is a waiting call, the vehicle is immediately dispatched and the call removed from the queue.

Because the effectiveness of an emergency operation is so dependent on vehicle (hardware) availability the failure/repair mechanism must be included in the model. Since vehicle failure may occur during operation or during operationally ready but idle periods, the failure mechanism is based on calendar time. Also since all vehicles in the service are assumed identical a single failure probability density function (pdf) is chosen which includes all vehicles. When a failure occurs the vehicle is removed from the ready pool and placed in a maintenance queue. The queue is examined at this point. If it is empty and maintenance resources (manpower, facility, and equipment) are available, the vehicle is placed in the repair facility. If the queue is not empty, the vehicle will be placed in it preserving a first-in-first-out order.

In many cases failure will occur during vehicle operation. To simplify the model it is assumed that none of these failures are catastrophic enough to render completion of the mission impossible. As a consequence of this assumption, when a vehicle failure occurs during operation, the vehicle is flagged, and upon completion of the service call is placed in the maintenance queue as described earlier. Normally a vehicle returning from a service call will be placed in the ready pool.

The maintenance queue is most frequently ranked on a first-in-first-out basis. Each repair action requires one maintenance resource set which is obtained from a pool of resource sets. Each set contains the manpower, repair tools, and facilities necessary to complete a repair action. For simplicity, all sets are considered identical and each failure requires one resource set to effect repair. Upon completion of a repair action, the maintenance resource set is returned to the resource pool: at the same time the vehicle is returned to the ready pool. The maintenance queue is then queried and, if not empty, another vehicle is removed from the queue and brought in for service.

Since most emergency vehicles are complex from a hardware point of view, regularly performed periodic maintenance is critical to effective system performance. The maintenance intervals are generally a function of usage, such as miles or flight hours. This important aspect of system operation is included in the model by maintaining and observing utilization on each vehicle in the system. At the conclusion of each service call the vehicle is interrogated to ascertain whether a periodic is due. If one is due, the vehicle is placed in the maintenance queue as described earlier.

An important management policy is included at this point. If an emergency call arrives and no vehicles are available (in the ready pool) the maintenance queue will be scanned to determine if a vehicle is waiting for periodic maintenance. Where required, these waiting vehicles will be dispatched to handle the emergency service calls and then returned to the maintenance queue at completion of the call.

The overall system operating logic described above is schematically depicted in Figure 1. In general, wherever management policies are built into the model and not available as input parameters to the user, the policy which provides maximum effective reaction to service calls was utilized.

The above logic structure conforms to that required for event type simulation. It has been programmed, debugged, and used as an instructional aid. The model runs on an IBM 1130 computer system with 8K words of storage and a disk drive (512K words). It utilizes the GASP II Simulation Language and the LOCAL feature available in the IBM 1130 software. For the reader who is not familiar with them, GASP and the IBM 1130 system will be briefly reviewed below.

The GASP Simulation Language was designed to provide the small computer user with simulation capability. The basic system provides a series of FORTRAN subroutines which accomplish the following tasks:

a. Executive Control - including maintenance of the time file and event scheduling.

b. Set Manipulation - including filing, removing and finding items in sets or queues called files.

c. Data Collection - including collection of time and event based (discrete) data.

d. Input/Output - including simulation model initialization input, and output of standardized GASP collected data.

e. Function Generations - including the generation of random variables from standardized pdf's such as Normal, Erlang, Poisson, and Lognormal.

f. Error Tracing - including the determination and identification of functional errors.

The techniques employed in the GASP systems are basically similar to those found in the SIMSCRIPT languages. The user must first visualize or structure the system to be modeled in terms of events, entities, attributes, and sets. Because of the reduced size of most computers which run GASP, many simscript functions could not be included in the language, which requires the user to structure them in FORTRAN for his specific problem.

The basic approach to programming a GASP simulation requires that each event be a separate FORTRAN subroutine similar to EVENT Routines in SIMSCRIPT. A very simple mainline program is written which provides linkage to the GASP Executive System. The chronology of a GASP program's execution is as follows:

1. The user written mainline calls the GASP Executive.

2. The GASP Executive causes initialization of the simulation model through data input and obtains the first event identification.

3. Control is transferred to a user written subroutine (called EVNTS) which calls the appropriate event subroutine.

4. Upon completion of that event processing, control is returned to the GASP Executive, which repeats steps 2 and 3 until the end of simulation is indicated.

The IBM 1130 Disk Monitor Operating System provides an automatic overlap procedure to conserve core storage. The user specifies those subroutines which can be stored on the disk and loaded on-call (LOCAL Subroutines). Coupling this feature with the modular construction of the GASP system allows the simulation model to exceed core limitations many times. The penalty incurred in the use of LOCAL subroutines is a considerable increase in CPU operating time.

Returning to the discussion of the Maintainability Engineering Simulation System, the LOCAL feature and the modular construction of the GASP system are used to good advantage. All the event routines are made LOCAL's, and as a consequence, this model is able to run on an 8K machine. The four primary events are described below:

1. CALL event - this event simulates the arrival of one emergency call, dispatches a vehicle or places the call in the waiting queue, and schedules the next call arrival.

2. RETRN event - this event simulates the return of a vehicle from a service call. If the vehicle has not experienced a failure; no calls are waiting; or a periodic maintenance is not required; the vehicle is placed in the ready pool. Otherwise, appropriate actions are initiated.

3. FAILR event - this event simulates a failure by determining the location of the vehicle (ready pool, dispatched, maintenance pool for periodic) and either placing it in the maintenance queue; or if the vehicle has been dispatched, it is simply flagged for appropriate entry to the maintenance facility upon return.

4. REPAR event - this event simulates the completion of a repair action and return of the vehicle to the ready pool. If other vehicles are awaiting maintenance, one is placed in the maintenance facility. There are several other subroutines which provide data input and end of simulation output which will not be described here. The next section discusses user input to, and output from, the simulation model.

Section III

Model Input/Output

The GASP Simulation Language provides features for standardized input and output. The standardized input initializes GASP system variables in a manner similar to the Simscript languages. User variables are initialized using Fortran I/O statements in the mainline program or a user written event routine programmed expressly for this purpose. Simulation output comes from standard GASP routines which provide information on the GASP variables and from user written (Fortran) output subroutines.

The data elements which must be input during any particular simulation run are listed in Figure 2. While the data formats are quite involved and will not be described here, the amount required for a run can be placed on less than 25 data cards. When several runs are required for a specific system, the input data changes usually result in removing and replacing only two or three cards.

Output from the Maintainability Engineering Simulation System can best be explained by describing the data as they appear in an output listing. Figures 3, 4, and 5 show the results of one run of the model. Figures 3 and 4 are standard GASP Summary reports and Figure 5 is a report written expressly for this model.

The report first provides descriptor information regarding the run, such as project number, date, user, etc. Below that is a list of the values for each parameter set. These are input to the model and describe the probability density functions used. The next two sections, **GENERATED DATA** and **TIME GENERATED DATA**, provide the most important output. Generated Data means data accumulated on the basis of discrete occurrences. For this particular model an example would be Generated Data code 1 which describes the number of calls that were not serviced immediately. The observations column contains the number of calls that waited; the MEAN, STD. DEV., MIN., and MAX. columns describe how long the calls (that waited) had to wait.

Time Generated Data means data accumulated over time. In this example, Time Generated Data code 1 provides the mean, standard deviations, maximum, and minimum number of vehicles in the ready pool over the time period shown in the column labeled total time. Similarly, Time Generated Data codes 3, 4, and 5 describe respectively the number of vehicles in the repair queue, the number of maintenance resource sets not in use, and the number of calls waiting for service over the times shown in the TOTAL TIME column.

Next on the output listing are summary statistics on each of the files used in this model. This data is identical, in method of collection, to the Time Generated data described above. Each time an entry is made in a file the GASP system collects appropriate statistics regarding the number in the file at that point in simulated time. In GASP, File No. 1 is always the time file and contains all events that have been scheduled to occur in the future. Files No. 2, 3, and 4 are the ready pool, repair queue, and waiting calls queue respectively.

The last page of output (Figure 5) provides availability and utilization data on each vehicle in the simulation. Included with this is the cummulative number of spare parts used to that point in simulated time. Below this is a Cross-Reference which summarizes the meaning with respect to this model of the various data elements provided in the GASP Summary Report.

As will be noted in reviewing the output, many different kinds of data are provided in the standard GASP output. It will also be noted that this information is presented with the very minimal explanative labeling. The reason for this is the small amount of core available to the simulation, and therefore the need to minimize that core required for nice to have, but not essential considerations.

In summary, the Maintainability Engineering Simulation System provides the user a comprehensive compilation of statistics regarding system performance and sub-system utilization and effectiveness. Parameters such as number of calls that wait for service and the time they must wait establish overall system performance. While statistics such as average number of maintenance resource sets idle over time; average number of vehicles operationally ready over time; the up-time, down-time, and availability of individual vehicles in the system; and the number of spare parts sets required during simulated time provide data on the operational effectiveness of the vehicle and maintenance sub-systems.

The next section describes the use of the Maintainability Engineering Simulation System as an educational aid.

Section IV

Use of the Model as an Instructional Aid

As indicated in Section I, the Maintainability Engineering Simulation System was primarily intended as a teaching aid. The function of systems engineering (particularly maintainability and reliability) in hardware development is to assure that an end-item is produced, which meets customer performance and operational requirements, at the lowest possible cost. An important aspect of this cost is the so-called "cost of ownership" or "support cost". The fact that design measurably affects this cost is well known, but the way it affects cost is not. In training systems engineering personnel, this model provides the mechanism which demonstrates, and allows experimentation with, the interactions among design, support, operation, and cost parameters.

After the student has been given sufficient lecture to familiarize him with the subject, he is assigned the problem of developing an emergency system. For example, the following is a problem statement used to start a student project to develop an ambulance system:

Development Requirement for an Ambulance System

Problem: Develop an ambulance system to service four cities, each of which experiences the following historical requirement for services:

1. The time between the arrival of calls follows a negative exponential distribution with

   a. Mean - .1 hour
   b. Minimum - .05 hours
   c. Maximum - 5.0 hours

2. The time-on-call distribution is normal with

   a. Mean - 1.5 hours
   b. Maximum - 4.5 hours
   c. Minimum - .5 hours
   d. Std Dev - 0.9 hours

3. The miles-traveled-per-call distribution is normal with

   a. Mean - 35 miles
   b. Maximum - 100 miles
   c. Minimum - 10 miles
   d. Std Dev - 30 miles

These four cities have combined their resources to more effectively develop an ambulance service and have hired you, as an expert consultant in the field of maintainability, to tell them what type of system they should procure. Since the cities are very similar you will only investigate one, recognizing that the system would be identical for the others. Your cost data should be for all four cities, though. The development cost, of course, is one time and should be applied only once, not for each city.

You will have to develop this system under the constraints established in the tables included in this problem. The objective is to achieve the lowest practical life cycle cost while maintaining a desired level-of-service criterion. The simulation model described herein will provide a useful analytic tool in arriving at a final configuration.

You will specify the configuration by stating the mean-time-between-failure, mean-time-to-repair, preventative maintenance period, mean-preventative-maintenance-time, miles before overhaul, number of vehicles required, and number of maintenance men per shift required. The service level criterion you select for the system must be stated and justified in your report. You are free to utilize the simulation model as many times as is necessary to arrive at your configuration.

Parameters that can be changed by you in reaching the best solution include:

a. Number of vehicles in pool.
b. Number of maintenance men and skill level.
c. Preventative maintenance period. (in miles)
d. Preventative maintenance time distribution.
e. Corrective maintenance time distribution.
f. Reliability of the vehicles.

Given parameters: (provided by instructor)

a. Time between calls.
b. Time distribution of service calls.
c. Distance distribution of service calls.
NOTE: These parameters describe demand.

NOTE: Parameter changes affect cost factors as shown by relationships included in Tables 1-8.

As implied in the above problem statement, the instructor sets the parameters governing the demands for the system. These include the time-between-arrival of calls, the time required to service each call, and the number of miles traveled for each call. The student is free to exercise the model, changing parameters as he desires, in an attempt to find the hardware/support configuration which provides the most appropriate level of service at the lowest life cycle cost. He is, of course, constrained by development costs and operating costs. For example, by designing a system with lower repair time for corrective maintenance, vehicle availability will naturally increase. The development cost will also increase as a result. But the chief method of reducing repair time is to design-in greater modularity. This tends to increase the number of functions accomplished by each replaceable unit. This also reduces fault detection, isolation, and remove/replace times. Unfortunately the greater complexity increases measureably the cost of spare parts and therefore it increases the life cycle cost.

The instructor provides the student with curves describing the cost implications of various design and support parameter combinations. For the problem statement included above, the functions relating parameters values to cost are shown in Tables 1-8. These costs are related in a simplified Life Cycle Cost equation provided with the problem statement. This equation is shown in Figure 6. The instructor can measureably change this problem through changes in these cost relationships.

At this point a brief discussion of the solution the student is required to obtain is in order. As pointed out in the problem statement, the student must decide on, and justify, the criterion indicative of a good system. Most hardware designers would tend to develop greater system availability, using availability as the primary indicator for system worth. In many cases, a very high availability can be obtained without providing adequate service. Therefore, the primary criterion of goodness becomes the number of calls that must wait for response and the duration of this wait. And, for the problem used as an example here, the most appropriate solution (best response and lowest life cycle cost) results in a much lower vehicle availability than could have been achieved: a dichotomous result for a system engineering student whose education and training stresses more effective systems, with higher availability, through design. A valuable lesson learned: that suboptimizing the system by optimizing vehicle effectiveness can in fact degrade overall system performance.

Section V

Conclusion

The Maintainability Engineering Simulation System described in this paper is an application of discrete (event type) simulation to education. The model provides the instructor of system engineering (particularly reliability and maintainability) disciplines with a valuable teaching aid. This aid dynamically conveys to the student the important interactions between design, support, operation, and cost parameters. It allows the student to develop a system, in its entirety, through experimentation and analysis.

The more advanced student may use statistical and analytical techniques for suboptimizing the various parameters. He may use sensitivity analysis to isolate significant parameters. If the instructor desires, the simulation model can be run for a less than optimal configuration with a log-type output describing each event as it occurs (i.e., a call arrival, repair initiation, a failure, a return from a service call, etc.). Using this log as a history data the advanced student can be asked to summarize the data, fit standardized probability density functions to the data using statistical procedures, and use the results to provide initial forecasts for operation requirements.

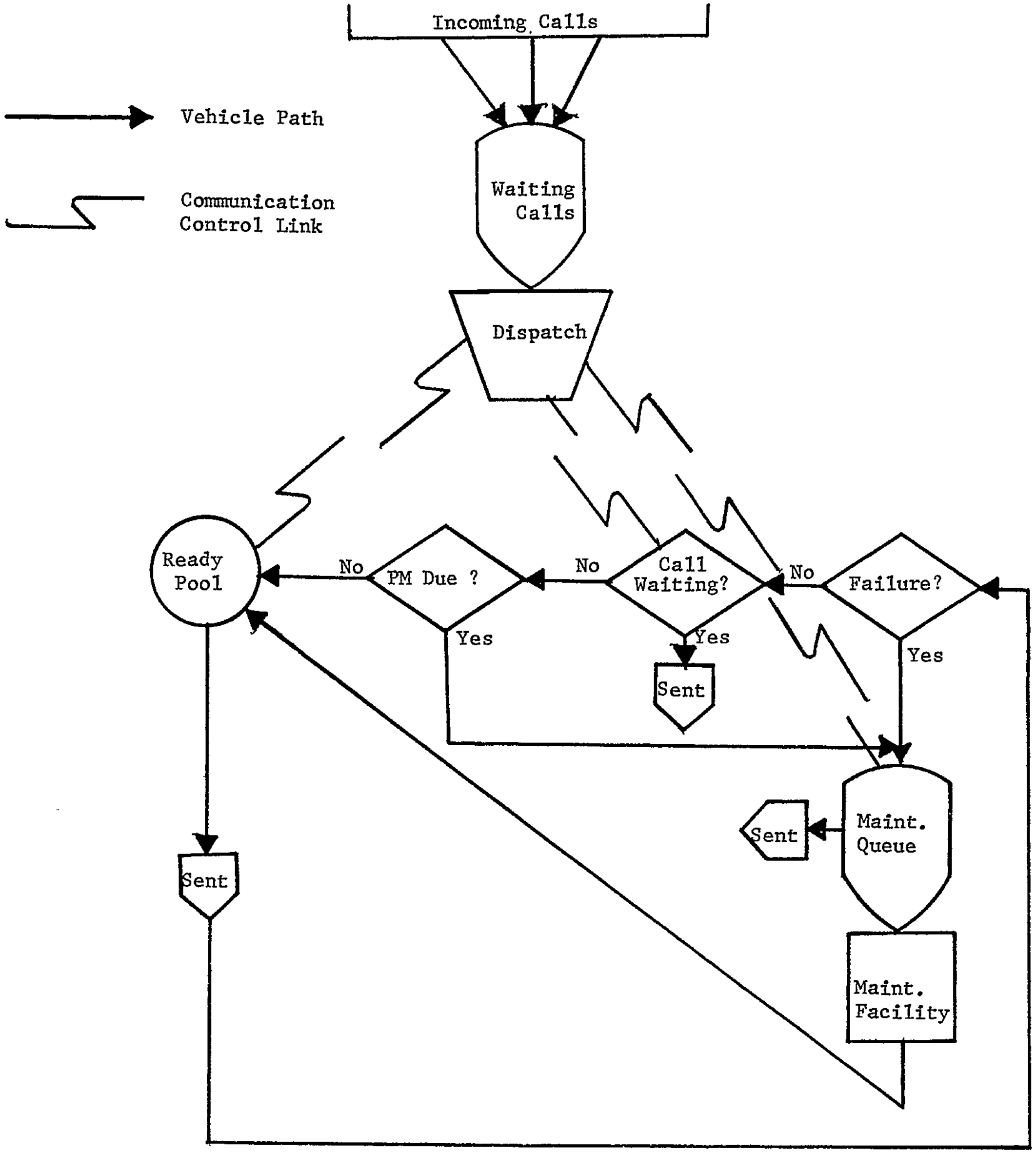
A secondary benefit derived from use of the model is an understanding, gained by the student, of the techniques of digital simulation. Experience gained to date has shown that, as the student becomes involved in the project, he tends to question the procedures and logic of the model. This naturally stimulates learning of the logic of the simulation process.

In conclusion, the Maintainability Engineering Simulation System:

a. is an important instructional aid for training systems engineers,

b. can be used to increase student knowledge of simulation, and

c. demonstrates the simulation capability possible with small core computer systems.



General Logic

Figure 1

CATEGORY	DATA INPUT ITEMS
Operation Parameters -	pdf for time between calls pdf for duration of each call pdf for miles per call* number of vehicles for this site.
Vehicle Parameters -	pdf for time before failure pdf for time to repair failure (corrective maintenance) pdf for time to complete periodic maintenance miles before periodic*
Support Parameters -	number of maintenance resource sets
Management Policy -	maintenance queue which can be ranked high value first or low value first on either time of entry, vehicle I.D. number, or maintenance code (periodic or corrective) call wait queue ranked HVF or LVF on times of entry into queue
Simulation Parameters-	Users Name Date Project Number Number of runs Time between runs Time of end of simulation Random number generator starting point

*NOTE: Could be operating hours, rounds fired, etc.

Input Data Items

Figure 2

```
                    **GASP SUMMARY REPORT**

          SIMULATION PROJECT NO.    5  BY   R. MORRIS

          DATE   3/ 22/ 1971              RUN NUMBER    5

PARAMETER NO.     1       1.5000      0.5000          4.5000          0.9000
PARAMETER NO.     2      35.0000     10.0000        100.0000         30.0000
PARAMETER NO.     3      -1.0397 -99999.0158          1.3860          0.8320
PARAMETER NO.     4       1.0000      0.50000         5.0000          0.2500
PARAMETER NO.     5       5.0000      0.50000        50.0000          2.5000
PARAMETER NO.     6      60.0000     10.0000         80.0000         30.0000


                           **GENERATED DATA**
       CODE       MEAN        STD.DEV.       MIN.         MAX.        OBS.

        1        0.4099        0.2337       0.1999       0.8999        20

                           **TIME GENERATED DATA**
       CODE       MEAN        STD.DEV.       MIN.         MAX.      TOTAL TIME

        1        5.6619        2.0913       0.0000      10.0000     1199.9001
        2             NO VALUES RECORDED
        3        0.0969        0.3664       0.0000       4.0000     1198.0002
        4        1.1369        0.7900       0.0000       2.0000     1197.5002
        5        0.0054        0.1009       0.0000       3.0000     1175.3000


                 FILE PRINTOUT, FILE   NO.   1

             AVERAGE NUMBER IN FILE  WAS    14.2622
             STD. DEV.                       1.9469
             MAXIMUM                           22

                       FILE CONTENTS                          FILE PRINTOUT, FILE   NO.   2

1200.4001      3.0000      0.0000      0.0000             AVERAGE NUMBER IN FILE WAS        5.6948
1200.9001      4.0000      9.0000   1198.0002             STD. DEV.                         2.1027
1201.7001      4.0000      1.0000   1199.5002             MAXIMUM                             10
1201.7001      4.0000      5.0000   1199.7001
1202.3000      1.0000      3.0000      2.0000                        FILE CONTENTS
1203.2001      4.0000      3.0000   1199.8000
1204.9001      2.0000      6.0000      2.0000
1205.1001      2.0000     10.0000      2.0000
1213.8000      1.0000      9.0000      2.0000
1220.0002      1.0000      1.0000      2.0000
1235.2001      1.0000      4.0000      2.0000
1237.2001      1.0000      2.0000      2.0000
1243.5002      1.0000      7.0000      2.0000
1258.2001      1.0000      5.0000      2.0000
```

Figure 3 - End of Run Report (Page 1)

```
             MAXIMUM                            10

                         FILE CONTENTS                        FILE PRINTOUT, FILE   NO.   4

1197.6001      0.0000      3.0000      0.0000             AVERAGE NUMBER IN FILE WAS        0.0120
1197.8000      0.0000     10.0000      0.0000             STD. DEV.                         0.1584
1199.3000      0.0000      6.0000      0.0000             MAXIMUM                              5
1199.3000      0.0000      1.0000      0.0000
                FILE PRINTOUT, FILE   NO.   3                        THE FILE IS EMPTY

            AVERAGE NUMBER IN FILE WAS        0.0000
            STD. DEV.                         0.0091
            MAXIMUM                             1

                 THE FILE IS EMPTY
```

Figure 4 - End of Run Report (Page 2)

```
                          END OF RUN REPORTS

                               TIME PERIOD     960.00  TO  1200.00


VEHICLE ID         OPERATIONAL          IDLE        DOWNTIME       AVAILABILITY        MILES


     1                427.79           642.09        130.10          0.89157           10485
     2                436.79           645.39        117.80          0.90182           10458
     3                439.69           637.29        123.00          0.89749           10719
     4                447.29           644.59        108.10          0.90991           10689
     5                426.59           662.39        111.00          0.90749           10799
     6                437.49           643.39        119.10          0.90074           10404
     7                430.39           638.99        130.60          0.89115            9989
     8                405.79           637.39        156.80          0.86932            9917
     9                401.29           654.49        144.20          0.87982           10319
    10                427.29           626.49        146.20          0.87815            9606

TOTAL                4280.46          6432.55       1286.98          0.89275          103385.0


                  TOTAL SPARE PARTS TO DATE      369



                              CROSS-REFERENCE


            TIME GENERATED DATA                         FILES

SET                    STATISTIC                         NO.                 CONTENTS

 1                     VEHICLES AVAILABLE                 1                  TIME FILE
 3                     REPAIR QUEUE TIMES                 2                  AVAILABLE POOL
 4                     MANPOWER UTILIZATION               3                  REPAIR QUEUE
 5                     CALL WAIT STATISTIC                4                  WAITING CALLS

                     CALL WAIT TIME - GENERATED DATA SET 1
```

Figure 5 - End of Run Report (Page 3)

$C_T = C_D + C_P + C_O + C_{OV} + C_R + C_{DR}$ where

C_D = Cost Development

C_P = Cost of procurement (initial acquisition)

C_O = Cost of operation:

a. Maintenance Labor Cost

b. Spare Part Cost

C_{OV} = Cost of overhauls

C_R = Cost of replacing wornout vehicles

C_{DR} = Cost of drivers (one driver per vehicle)

C_T = Total cost of procurement and operation - assume 5 year period for this problem and give total and yearly cost.

Cost Equation

Figure 6

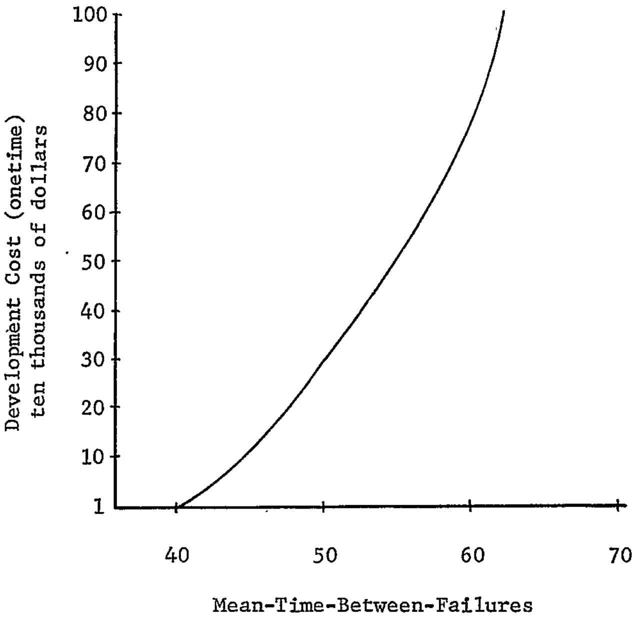


Table 1--Development Cost vs Reliability

NOTE: Standard deviations remain at a constant ratio with mean-time-to-failure such that the standard deviation is approximately 1/2 the mean life.

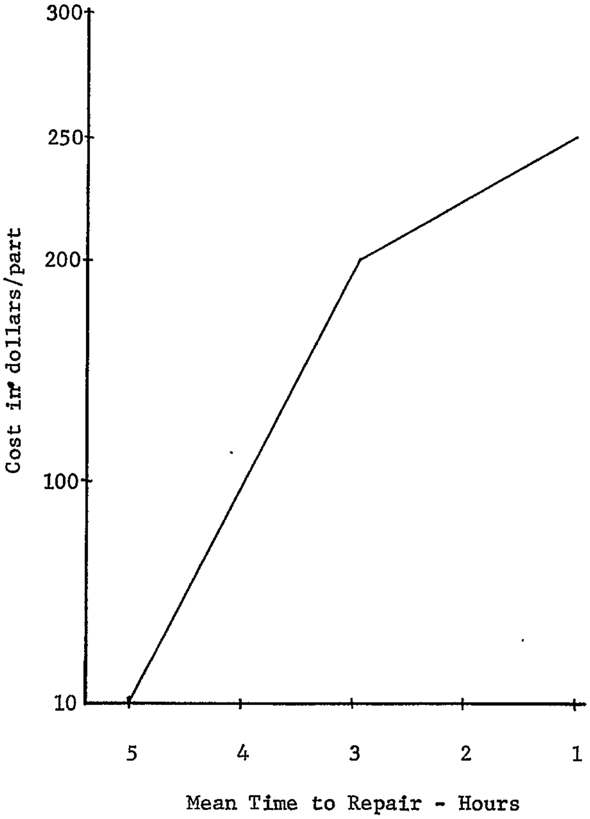


Table 3--Cost of Spare Parts vs Mean-Time-To-Repair

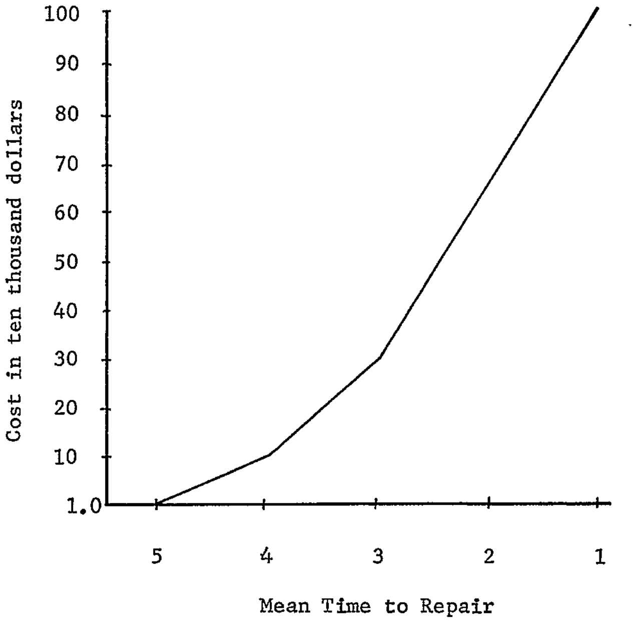


Table 2--Development Cost vs Repair Time For Corrective Maintenance

NOTE: The decreases in repair time indicated are achieved primarily through modular replacement. When repair time reaches less than 2 hours mean-time-to-repair the system does not require overhaul as all failure prone components are included in modules. Naturally, as modularization increases part cost increases, see Table 3. Also see Table 6, for reduction due to increased skill level. Standard deviation for repair times should remain at approximately 1/2 the mean repair time.

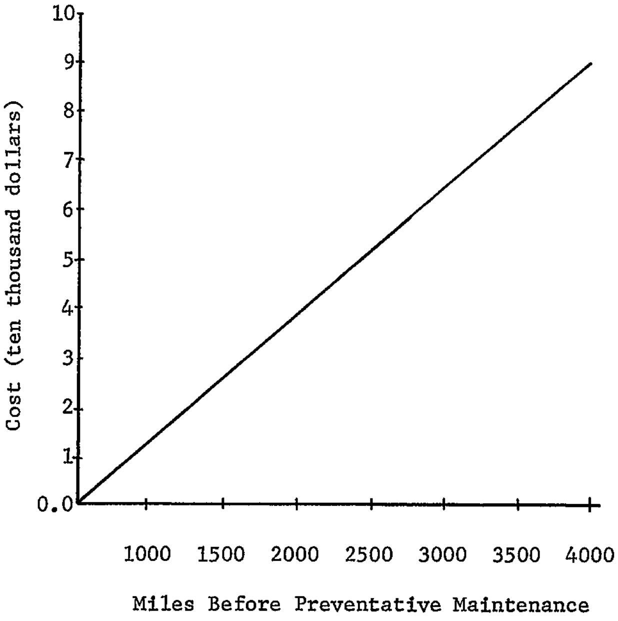


Table 4--Development Cost vs Preventative Maintenance Period

NOTE: See Table 6 for possible reduction due to skill level improvements.

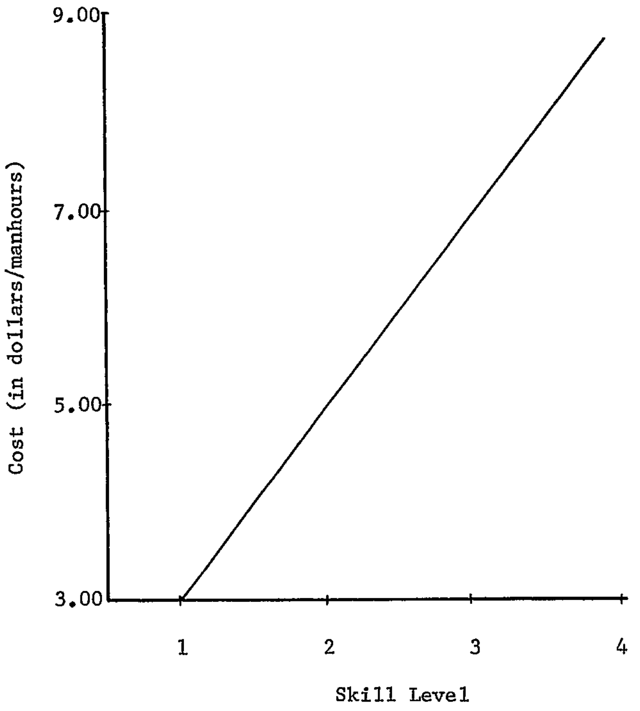


Table 5--Skill Level vs Labor Cost

NOTE: This labor rate includes overhead, facilities, and administration. Since skill level has a definite affect on repair time see Table 6.

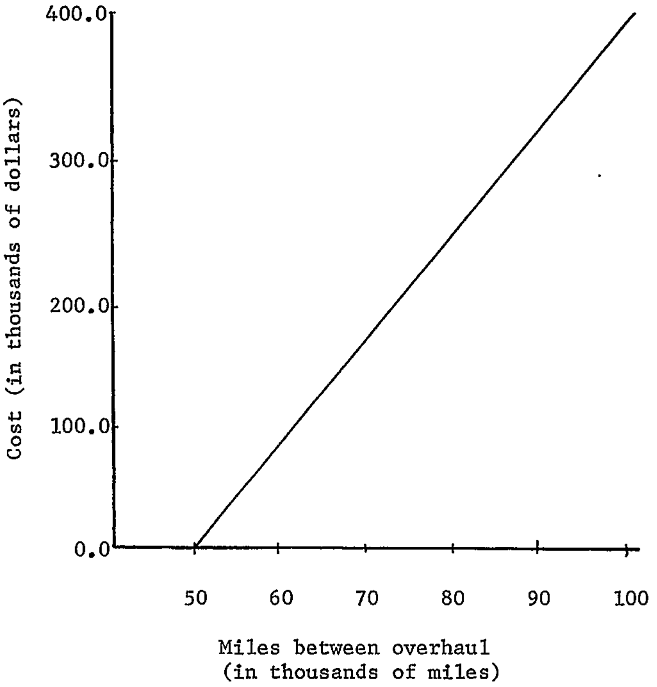


Table 6--Development Cost vs Miles Between Overhaul

NOTE: The maximum number of overhauls before vehicle replacement is two. Cost per overhaul can be estimated from the following equation:

Cost/overhaul=1000+(mean time to repair)1000

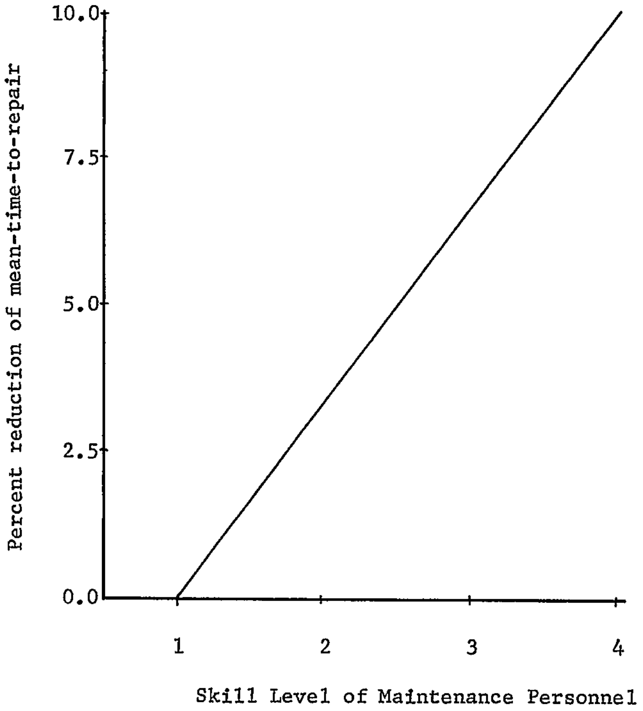


Table 7--Maintenance Time Reduction vs Skill Level

NOTE: This table is used with Tables 2 and 4 to determine repair times and with Table 5 to determine cost.

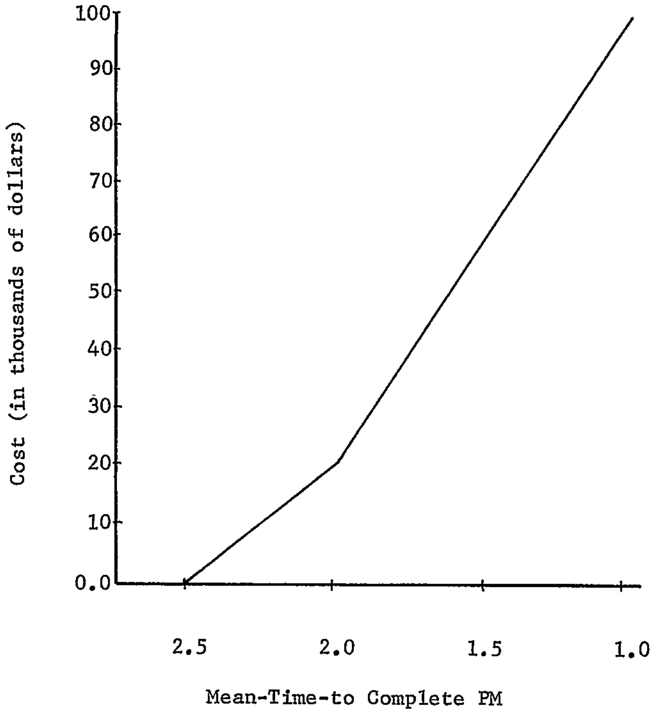


Table 8--Development Cost vs PM Time

NOTE: Standard Deviation of PM time distributions should be approximately 1/4th the mean time.