Unclassified
SECURITY CLASSIFICATION OF THIS PAGE

DTIC FILE COPY

1

AD-A191 000

OCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1b. RESTRICTIVE MARKINGS

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited

4. PERFORMING ORGANIZATION REPORT NUMBER(S)

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Air Force Operational Test and
Evaluation Center

6b. OFFICE SYMBOL (If applicable)
HQ AFOTEC/LG4

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
HQ AFOTEC/LG4
Kirtland AFB NM 87117-7001

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION

8b. OFFICE SYMBOL (If applicable)

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)

T-46A Availability Model (U)

12. PERSONAL AUTHOR(S)
Sarah J. Fritz, Capt, USAF

13a. TYPE OF REPORT
Documentation

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (Year, Month, Day)
31 May 1987

15. PAGE COUNT
66

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
04	14	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
~~Availability, Simulation, SLAM,~~ Operational Suitability, ~~IOT&E,~~ Test Aircraft, T-46, Model, Analysis

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The T-46A Availability model is an analysis tool to aid in the operational suitability evaluation of the T-46A during initial operational test and evaluation. ~~(IOT&E)~~ The model was intended to be used to estimate the fully mission capable (FMC) rate, the maintenance man-hours per flying hour (MMH/FH), and the mean downtime per sortie (MDT/S). The simulation model is a means to use the information gathered during IOT&E combined with information on the ATC environment to evaluate how the system will perform in its intended environment. The parameters which can be measured or evaluated during test are task times, failure rates, reliability growth, number and specialty code people doing the maintenance work. The items which are "simulated" are number of aircraft, number of sorties, weather cancellations, and maintenance personnel available each shift. The model was validated using T-37 inputs and comparing the model output to actual field performance of the T-37. The model is programmed using Simulation Language for Alternative Modeling (SLAM).

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
BURTON E. McKENZIE, Lt Col, USAF

22b. TELEPHONE (Include Area Code)
(505) 844-0346

22c. OFFICE SYMBOL
HQ AFOTEC/LG4

Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

T46A AVAILABILITY MODEL

Captain Sarah Gjerstad Fritz
HQ AFOTEC/LG4A
Kirtland AFB, NM 87117

January 1988

TABLE OF CONTENTS

INTRODUCTION

PURPOSE

The T46A Availability Model is an analysis tool to aid in the operational suitability evaluation of the T46A during initial operational test and evaluation (IOT&E). The model will be used to assess the fully mission capable (FMC) rate, the maintenance manhours per flying hour (MMH/FH), and the mean downtime per sortie (MDT/S).

The FMC rate is the measure of effectiveness used to evaluate the availability objective (1:B-1). Availability is a measure of the degree to which an item is in an operable and committable state when the mission is called for at a random point in time (3:8). The FMC rate is the percentage of possessed time that a system is capable of performing all of its assigned missions (3:8).

The MMH/FH and MDT/S are the measures of effectiveness used to evaluate the maintainability objective (1:B-9). Maintainability is the measure of the capability of an item to be retained in or restored to a specified condition when maintenance is performed by personnel having specified skill levels, using prescribed procedures and resources at each level of maintenance and repair (3:8). MMH/FH is the total base-level direct maintenance manhours on- and off-equipment, required to support the system, divided by flying hours. MDT/S is the total time the system is not mission capable for maintenance (NMCM) (including maintenance delay times), partial mission capable for maintenance (PMCM), or partial mission capable both for maintenance and supply (PMCB), scheduled or unscheduled, in clock hours, divided by the number of sorties (3:10).

WHY USE A MODEL

The HQ ATC requirements for FMC, MMH/FH and MDT/S are based on a mature system operated at an ATC base. Table 1 summarizes the differences between the field environment and the test environment. Because of these differences FMC, MMH/FH, and MDT/S measured during test would not be an accurate assessment of the T46A performance in the field.

Although the environments are different, there are parameters which can be measured or evaluated during test - task times, failure rates, reliability growth, number and specialty code of people doing the maintenance work, etc.

TEST ENVIRONMENT	OPERATIONAL ENVIRONMENT
1-4 aircraft	about 100 aircraft per location
evolving system	mature system
test flying schedule	operational flying schedule
test pilots	student pilots
maintenance personnel	maintenance personnel
contractor/AF 7-level	AF 3 and 5-level
unconstrained quantity	constrained quantity
contractor supply system	AF supply support
Edwards AFB	ATC bases

Table 1. Environmental Differences

A simulation model is a means to use the information gathered during test combined with information on the ATC environment to evaluate how a system will perform in its intended environment. The information on the ATC environment is used to "simulate" this environment. Those items which are simulated are: number of aircraft, weather cancelations, utilization rate, periodic inspections on airframe and engines, and maintenance personnel available each shift.

BACKGROUND

SYSTEM DESCRIPTION

The T-46A is a twin-engine aircraft with side-by-side seating. The cockpit is pressurized and contains modern avionics and instrumentation. Aircraft range and endurance design performance allows a formation training mission of 1.5 hours, followed by an instrument approach at home base, and then a 300-nautical-mile (NM) divert to an alternate airfield with Air Force Regulation 60-16 fuel reserves. The T-46A aircraft will contain anti-ice equipment for climb and descent through moderate icing conditions. The engine noise will meet the EPA T-1 turbine-engine noise standards. (1:3)

OPERATIONAL CONCEPT

As the UPT primary phase trainer, the T-46A will be used in contact, instrument, navigation, and formation flying. It will also be used for low-level, tactical navigation training in the undergraduate navigator training (UNT) program and for pilot maturation in the Strategic Air Command's Accelerated Copilot Enrichment (ACE) program. (1:3)

MAINTENANCE CONCEPT

ATC and AFLC will use organizational, intermediate, and depot level maintenance to maintain the T-46A system. Specific levels of repair for subsystems, line replaceable units (LRUs), and shop replaceable units (SRUs) will be determined by repair level analysis. Equipment, systems, and components which are presently established in the Air Force and DOD inventory and used on T-46A systems will be maintained according to existing maintenance and support concepts. Air Force personnel will perform organizational and intermediate level maintenance, except at Vance AFB, which has contract maintenance. (1:6)

SCENARIO EVALUATED

All quantities discussed in this section are inputs and can be changed in the data file. Most of the inputs are what can be expected at Laughlin AFB, TX. This base was used because it will be the first base to get the T-46's and because for many factors it could be considered a typical or average base. Some of these factors are number of aircraft, UTE rate, and weather cancellations.

There are 108 aircraft, which includes 8 ACE aircraft. Each day 65 are scheduled to fly and 8 are scheduled as spares. These are the only aircraft which are preflighted; and if necessary, a sortie will be missed, rather than use another aircraft. The number of sorties scheduled each day varies due to student load, weather cancellations, and daylight hours.

The manpower available for the T-46 is the same as currently authorized for the T-37. The specialists work two eight hour shifts, five days a week; the crew chiefs work three shifts and weekends.

The analysis is for a mature system. This means the wing has all T-46's (no overlap with T-37's), has a typical mix of manpower skill levels, and the T-46 has reached MTBM maturity.

Figure 1 shows the macro level logic chart for the scenario being simulated. Each day the number of sorties to be flown that day is calculated. For each sortie, if there is an aircraft which is FMC, the sortie is flown. After the sortie, the aircraft is repaired (if there has been a failure) or refueled, and returned to FMC status to be flown later that day, if necessary. If the aircraft is not FMC, the sortie will be flown if it can be repaired within 30 minutes or if there is a spare aircraft available.

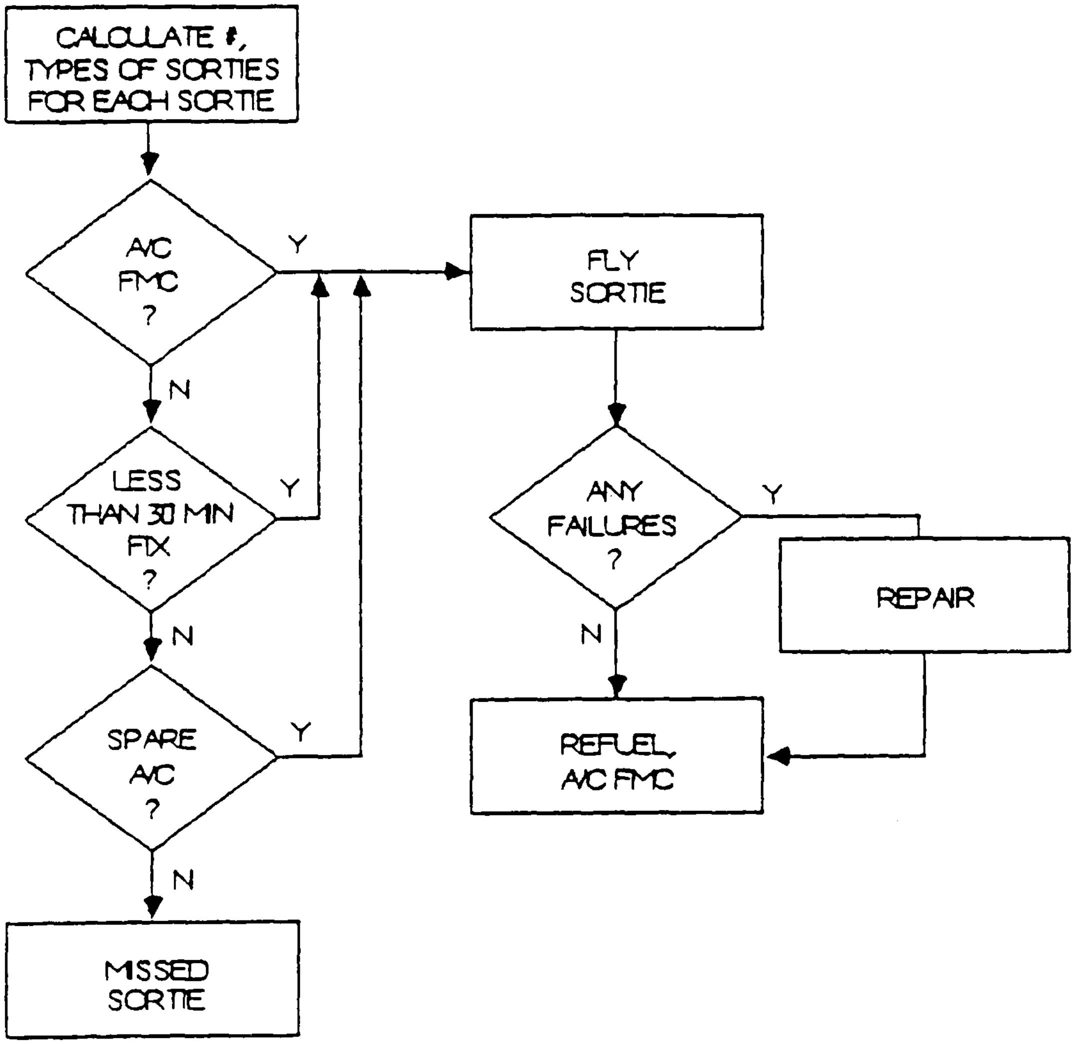


Figure 1. Macro Level Logic Chart

COMPUTER LANGUAGE USED

The T46A Availability Model was developed using the Simulation Language for Alternative Modeling (SLAM). SLAM is a special purpose FOR~~T~~RAN based simulation language which allows an event-scheduling and/or a process-interaction orientation toward modeling (2:99). The type of orientation one uses depends on the level of complexity needed to model the system and the extent to which the model will have to be embellished for future uses (4:315).

The event-scheduling orientation concentrates on events and how they affect the state of the system. This method uses a FORTRAN model to schedule events, and then processes the events at the scheduled time. FORTRAN subroutines are used to control the changes associated with each event, which may entail manipulating files, collecting statistics, and/or printing status reports (4:73). This is called a discrete-event model because changes in the model occur at discrete points in time.

The process-interaction approach concentrates on entities and the sequence of events and activities they undergo as they flow through the system. The processes are represented by the nodes and branches of a network. Consequently, a network model represents the processes that an entity goes through as it passes through the system (4:73).

The T46A Availability Model uses both the event-scheduling and the process-interaction approaches toward modeling. Because of the complexity of this model, the event-scheduling (FORTRAN) orientation is more extensive than the process-interaction (network) orientation.

The events are used for scheduling sorties, controlling the cal- endar, controlling the shift changes (including the number of resources available on each shift), determining the status of aircraft, routing aircraft to the repair network (after determining the task time and the resources needed for the repair), seizing and freeing resources, and calculating statistics. These events are described in more detail in the narrative description.

The network is used to represent the process an aircraft, engine, or LRU must undergo to be maintained.

NARRATIVE DESCRIPTION

The narrative description of the input file, the FORTRAN subroutines, and the network is intended to give an overall picture of what is being considered in the model. Refer to Appendix E - Run Instructions for more detailed information on the input file. Refer to Appendix B - Summary Logic Charts, Appendix C - List of Variables, and Appendix D - Computer Code for more information on the subroutines.

DATA FILE

The T46AD FORTRAN file contains all of the inputs for the model in a block data format. The type of inputs included are data file, scenario, system level, and WUC.

The data file inputs explain how to read the data file: # of resources, resource type used as dummy, # of WUC's, and WUC used for engine core. For those tasks which only use one resource type, a dummy resource is used. The WUC number for the engine core is needed to track total engines during engine phase.

The scenario inputs include information related to the operational environment. It includes # of aircraft, utilization rate, number of resources assigned, resource productivity factor, and schedule information by month.

The system level inputs include information on the flying hours between airframe and engine phase inspections. It also contains information the resource types, number of resources, and task time for system level tasks.

The WUC level inputs contain information by WUC of the MTBM, the probability of various task occuring. It also contains the resource types, number of resources, and task times to troubleshoot, repair in place, remove and replace, perform an operational check, and to repair the WUC in the shop. These inputs can be at any level of detail. In the current version, the 20 highest failure items are modeled at the LRU level and the other WUC's are at the 2-digit WUC where possible (in cases where different AFSC's work on the same WUC, it was modeled at a lower WUC).

FORTRAN SUBROUTINES

Except for INTLC, the subroutines are listed alphabetically, rather than in the order they are called. For each subroutine the description includes how the subroutine is called or scheduled, what it does, and what other subroutines it calls.

INTLC - This subroutine is called internally by SLAM at the beginning of each run. It initializes all counters, sets the initial number of resources (people and support equipment), calculates the annual number of sorties, sets failure clocks, creates airplanes and assigns them inspection times. INTLC initially schedules the following subroutines: FLYING - which starts the daily flying; FLYEND - which removed the planes from the flying schedule at the end of the day; CALNDR - which is the month counter; SHIFTS - which controls the people working on each shift; and STATS - which calculates and prints the output at the end of the run.

ASSGNO - this subroutine is called by CHECK3 at the start of each repair. For the item being repaired it generates a set of pseudo- random numbers and compares each of the following inputs to a random number: on aircraft repair, functional check, functional check flight, cnd, rejection after functional check, and tow for repair. Each input is the probability of that task occurring for the item being repaired.

CHECK1 - This subroutine is called at the end of each sortie and compares the flying hours of the aircraft to the flying hours before the next airframe and/or engine phase inspection. If no inspection is needed, CHECK3 is called to check for failures. If no failures, the aircraft is returned to the FMC queue after the turnaround time.

If an inspection is needed, it assigns a new inspection time, decides if its a major or minor phase, and if it will be painted; then the plane enters the phase network. For engine inspections, it assigns a code for the type of task and sends two engines to the task network. If the engines are not inspected, two spare engines are created, because at the completion of the airframe phase there is a demand for two engines.

CHECK2 - This subroutine is called from the network following a repair. It calls CHECK3, which checks for a second failure. If none, it checks if any items have been cannibalized and sends the plane to the repair network if necessary. If not, the repair is completed and the downtime for the current repair and the MDT/S are calculated. Finally, if the plane was needed for the sortie, the downtime is checked. If less than .5 hours, the sortie is flown (the plane is sent to the sortie queue); if more than .5 hours, the sortie is counted as a missed sortie.

CHECK3 - This subroutine is called to check for a failure before a sortie (from SORTIE), following a sortie (from CHECK1) and following a repair (from CHECK2). It compares the flying hours to the failure time for each of the WUCs or LRUs modeled. If the flying hours is greater than the failure time, there has been a failure and CHECK3 calls ASSGNO and sends the plane to the repair network. If CHECK3 is called before the sortie, the failure time is compared to the flying hours plus ten minutes (this accounts for about 10% of the failures found during preflight inspections).

CALNDR - This subroutine controls the counters for the day, week, and month. If it is a weekday, it also calculates how many sorties will be flown that day. The number of sorties is based on the amount of flying, number of flying days, weather cancellations, and any sorties missed the previous day. For the monthly counter, CALNDR is scheduled in INTLC initially identifies it as a monthly counter) and reschedules itself for each subsequent month. For the daily and weekly counters, it is called from FLYING each day. At the end of each week, it writes the week number to the screen.

FLYING - This subroutine is scheduled initially in INTLC and reschedules itself 24 hours later (once a day) each time it is called. It calls CALNDR, which calculates the number of sorties for that day and returns the number to be scheduled initially(one-eighth of the daily sorties). (Scheduling the sorties in groups thoughout the day reduces the number of events on the event calendar, which uses less space and decreases the run time). First, this subroutine removes the aircraft on the flying schedule and the spare aircraft from the HOLD queue and puts them in the RDY (ready) or SPR (spare) queue, which represents the preflight tasks. The subroutine then schedules the initial group of daily sorties (beginning 30 minutes prior to sunrise), the night sorties (beginning 30 minutes after sunset), the cross-country sorties (if Friday), and event 2 - FLYPM (which will schedule the remaining daily sorties.

FLYPM - This subroutine is scheduled in FLYING to occur seven times throughout the day. When called it schedules one-eighth of the sorties for that day.

FLYEND - This subroutine is scheduled initially in INTLC and reschedules itself for 24 hours later each time it is called. It increments the manhours because of postflight inspections and if necessary, increments the manhours because of 200FH scheduled maintenance or monthly maintenance. It removes the planes from the RDY (ready) and SPR (spare) queues.

MXSEIZ - This subroutine is called from the repair network. First it determines what resources are needed and how long the task will take. For engine phase, test cell, auto ecs, or tow, this information is taken from the TASKIN array; for shop tasks, from the SHOP array; for troubleshoot, repair-in-place, remove and replace, and functional check, from the PEOPLE and TIMES arrays for the LRU or WUC being repaired. The type and number of resources needed are coded as attributes (so they are identified with a specific aircraft) and as integer variables (to be used in the resource array).

MXSEIZ determines if the resources are available. For the remove and replace task, a supply delivery time is added. If available, the subroutine computes the time the task will be completed and compares this to the time the shift will end. If it will not be completed by the end of the shift, the task is broken into two segments - the time on current shift and the task time remaining for a future shift. At this point, the number of resources is decremented and the aircraft returns to the repair network. If the resources are not available, the aircraft is filed in a NMCS queue according to which resource is causing the delay. The aircraft returning to the network is coded as a dummy and destroyed.

MXFREE - This subroutine is called from the repair network following completion of a task or the end of a shift. It frees the resources, calculates manhours, and schedules MXQCK to check if anything is waiting for the resources which have just been released. It sets the task time for the current shift to the task time remaining and sets the task time remaining to zero. (In the network, if the task time for the current shift is zero, the task is completed).

MXQCK - This subroutine determines if any aircraft are waiting for resources which have just been released (called from MXFREE) or are now available because of a change in the shift (called from SHIFTS). It checks every aircraft in each of the NMCM queues for the current shift. If the resources are now available, the aircraft is sent to the repair network.

SHIFTS - This subroutine controls how many resources are available. First, it zeros out the number of resources available for the shift that has just ended and moves the aircraft to the set of NMCM queues which correspond to the next shift. The shift number changes. The number of resources working on this shift is set at the number assigned for that shift, but is decreased (based on a productivity factor for each resource) to represent the number working that day. Finally, SHIFTS determines when the shift will change again and reschedules itself for that time. It also sets a code to the end of the shift, which is used in MXSEIZ to determine whether the task will be completed during the shift.

SORTIE - This subroutine initiates each sortie by locating an aircraft, calling check3 (which checks for failures), incrementing the flying hours, number of sorties, and maintenance manhours (due to launch and recovery). It then files the aircraft in the sortie queue (which represents flying the sortie) and schedules CHECK1 to occur at the end of the sortie. When the sortie is initially scheduled, in FLYING or in FLYPM, the sortie length and the completion time are set.

SPAREA - This subroutine is called from TASKNX at the completion of a shop repair or at the end of engine work. It increments the number of spare parts available and decrements the current demand for that part.

SPAREB - This subroutine is called from the remove and replace network and from the phase network (following airframe phase there is a demand for two engines). It decreases the number of spare parts available and increments the current demand for that part. It also sets the maximum demand.

SPPLY1 - This subroutine is called from the repair network when a part is needed, but not available. It checks the NMCS queue to find an aircraft from which a part can be cannibalized. In order to cannibalize, the part must be one which would be cannibalized (as indicated in the input file) and that part must not be already missing on the aircraft in the NMCS queue. If the part is found, the aircraft in the NMCS queue is coded as now missing that part, and the part becomes available for the aircraft in the repair network. In the repair network, the remove and replace time is doubled to account for the time to cannibalize. If the part is not found, the aircraft undergoing repair is sent to the NMCS queue, and SPPLY2 is called to order the part.

SPPLY2 - This subroutine is scheduled when a part arrives or is called from SPPLY1 to order a part. If called when the part arrives, it locates the plane in the NMCS queue which is waiting for that part. The SLAM default is first in, first out if more than one plane is waiting for the part. Once it locates the plane it sends it to the repair network. Following the repair, CHECK2 will check if any other parts are missing; if so, it will return to the NMCS queue. If called to order a part, it codes the plane as missing the part and schedules an order time. To maintain an NMCS rate of 5%, the order time is based on the current NMCS rate.

STATSO - This subroutine is originally scheduled in INTLC at sunrise and reschedules itself to occur every day. It checks the FMC rate (called FMC1) at the beginning of the day. This rate is better than the FMC rate.

STATS1 - This subroutine counts the number of aircraft which are waiting to be preflighted at sunrise. This is a better measure of whether the number of people on the night shift is sufficient than the average number of planes waiting for preflight (it doesn't matter how long they wait, just so they're done by sunrise).

STATS - This subroutine is scheduled in INTLC to occur at time TTFIN (input in SLAM file). It prints the maximum demand for each LRU or WUC modeled to the SLAM LISTING file. It calculates ithe mean downtime per sortie (MDT/S), the maintenance manhours per flying hours (MMH/FH), and the fully mission capable (FMC) rate and prints this information to a separate output file.

TURN - This subroutine is scheduled in CHECK1 if there are no failures. The turnaround time is calculated, and the average number of aircraft being turned is calculated and included as part of the FMC aircraft.

TASKNX - This subroutine is called from the task network. It increments the code for the type of task and then determines what task will be done next.

T38 - This subroutine is used to account for the manhours used by the T-38 for those workcenters which would share resources with the T-46. It creates entities which need resources. The time is determined by the number of manhours per year used on the T-38 by each resource type as indicated in the input file. These entities are sent to the network and terminated after flowing though it.

SLAM NETWORKS

The SLAM network is a generic network for all types of maintenance - scheduled and unscheduled, and for all types of equipment being main tained - aircraft, engines, LRUs. The equipment is assigned codes in the FORTRAN subroutines to determine the resources, time, and tasks needed for the maintenance. The entity representing the equipment enters the network. The first step is to get the resources needed for the repair. If the resources are not available, an entity is filed in one of the NMCM queues, and the entity in the network is terminated. If the resources are available, the entity either flows through an activity representing a supply delivery time, or it is sent to the section of the network representing no spare part is available (NOSP node), or it moves directly to the activity representing the time for the current task. Following the task time activity, the resources are released, and the entity enters the FORTRAN subroutine TASKNX to determine if the maintenance action is complete or if there is another task to be performed. If there is another task, the entity is recoded for the subsequent task and re-enters the network at the beginning of the task.

If the entity is sent to the NOSP node, it goes to a FORTRAN subroutine which checks to see if it is possible to cannibalize another aircraft for the spare part (the part must be one that would be cannibalized and the part must be available on another aircraft which is in the NMCS queue). If it finds a part, the task time is doubled and the entity is routed to the activity representing the task time. If it cannot find a part, the entity is sent to a FORTRAN subroutine which orders the part, schedules its arrival time and files the entity in the NMCS queue. The entity returning to the network is terminated.

ASSUMPTIONS

There are two types of assumptions - data assumptions and structural assumptions.

DATA ASSUMPTIONS

The data assumptions are generally related to the scenario being evaluated. These are quantitative assumtions, are input in the data file and can easily be changed. The following lists the source used for the quantities used. The quantities are listed in Appendix E - Computer Code.

Data Element	Source
# of aircraft	Expected at Laughlin, HQ ATC/XPQ letter
# scheduled to fly	% of assigned currently used
# sc'eduled as spares	% of assigned currently used
UTE rate	ATC SOC, AFOTEC test plan
Type of missions(day,local)*	ATC Maintenance Summary, Jan 81 - Jun 85
% of flying by month	ATC Maintenance Summary, Jan 81 - Jun 85
% weather cancellations	ATC Maintenance Summary, Jan 81 - Jun 85
sunrise, sunset	estimated from # of daylight hours
sortie length*	ATC SOC
FCF length*	Laughling quality assurance pilot
% repairs failing FCF*	Laughlin maintenance summary
sortie cancellation time*	current ATC policy
3 minute take-off intervals*	current ATC policy
limit on # cannibalizations*	current ATC policy
airframe phase interval	Fairchild recommendation
engine phase interval	Garrett recommendation
# of each AFSC	current Laughlin manpower listing for T-37
productivity factor	ATC skill level standards, weighted by # at level from Laughlin manpower listing
# of manhours for T-38's	ATC message
MTBM's by WUC	Fairchild prediction, AAA report
overall MTBM	current MTBM, projected to maturity
system level tasks	ATC, test team
WUC inputs	test team

Table 2. Data Assumptions

* indicates those items which are input in the FORTRAN code, rather than the data file.

Because these assumptions are quantitative, it is possible to do sensitivity analysis with any assumption to determine how much effect input changes have on the outcome.

Many of these assumptions can be relaced with observations once more is known about the system. For example, the times for preflight and postflight inspections and thru-flight servicing were assumptions prior to testing, but have now been replaced with actual experience.

STRUCTURAL ASSUMPTIONS

Structural assumptions pertain to how the system operates and involve simplifications of reality. Because of these simplifications, there are limitations on what the model should be used for and what confidence should be placed in the results. Since these assumptions become an inherent part of the model logic, they may not be easy to change.

The model measures the FMC rate only. It assumes that if there is any failure on the plane, it is repaired as soon as possible. No attempt is made to fly the plane until it has been repaired. (This does not apply to delayed discrepencies, which are not part of the failure rate and are repaired when the plane undergoes its phase inspection).

The availability of manpower is random and evenly distributed throughout the year, not according to workload.

The NMCS rate is an input, and the model adjusts the number of spares and the depot resupply time to maintain this rate. The rate does not vary with seasonal fluctuations in the amount of flying.

The triangular distribution was used for task times.

To account for failures which are found in preflight, the model compares the current flying hours plus 10 minutes to the time of failure. This implies that if a failure occurs within the first 10 minutes of the sortie, the problem would have been detected before the sortie started. While this may not be true, it is a means to account for the failures which are found during the preflight inspection.

The UTE rate is the average flying hours/aircraft/month. This varies by month, with more flying done in the summer and fall and less in the winter and spring. The number of sorties on a daily basis varies with the weather cancellation rate.

The engine phase inspection is 900FH and the airframe phase inspection is 600FH. The model assumes that there will be spare engines for the airframe while the engines are undergoing phase.

The model allows for cannibalization of spare parts through an input in the data file. If a part is coded as a part that would be cannibalized, the model will cannibalize immediately if the part is not available and there is a NMCS aircraft with that part. This can lead to a higher cannibalization rate than would be seen in the field.

There is no priority on tasks. By default, the priority used is first-in, first-out. Also, all tasks stop at the end of the shift, and are resumed at the next shift, provided the manpower is available. There is no time penalty for stopping and starting a shift. There is no overtime.

VERIFICATION AND VALIDATION

The purpose of model validation is two-fold: first to produce a model that adequately represents true system behavior; second, to increase the credibility of the model (2:376). Validation involves verifying that the computer code performs as intended (verification) and determining that the model is an accurate representation of the real system (validation). Although these procedures are discussed separately below, both are performed simultaneously and are an integral part of model development.

VERIFICATION

The SLAM language has several monitor statements, summary, trace, and files, which were used to verify that the code is working correctly. The summary statement allows the output to be printed periodically (at selected intervals) throughout the run. From this it was possible to determine that the results were not constantly increasing or decreasing, that each observation was reasonable, and that the statistics were being calculated correctly. The trace statement traces every move for a period of time and was used to follow the entities through the network to ensure they were taking the correct paths. The files statement printed the composition of each file at certain points in time. It was used to verify that no entities were being inadvertently created and/or destroyed and that the attributes of each entity were not inadvertently being changed.

The SLAM LISTING file contains the standard SLAM output and was used to check current and maximum queue lengths, average wait times, the number of aircraft in each activity at the end of the simulation and the total aircraft which have passed through each activity.

All the input parameters were printed out at the end of the simu- lation to insure that they had not been inadvertently changed during the run.

Finally, the methods of programming, assumptions, capabilities, and limitations of the model were discussed throughly with an expert in the logistics modeling area (5).

VALIDATION

There are two methods of validation - objective and subjective. Objective involves collecting data on the system and statistically comparing this to data (output) from the model. Since the T-46 is not yet fielded, this was not possible. However, this comparison was done for the T-37. The purpose of this comparison was to demonstrate that the model output adequately reflected the model inputs. The inputs used were CY84 ATC UPT averages from the AFLC D056 data base, the ATC Maintenance Summary, and expert judgement from maintenance technicians at Laughlin AFB, TX. Because of the aggregrate nature of the input data bases, this comparison does not represent any specific base. The actual performance data was collected by base and weighted by the number of aircraft at the base to arrive at ATC UPT average values.

	Model Prediction	Actual Performance
FMC	.83	.82
MDT/S	7.0	6.9
MMH/FH	2.9	3.1

Table 3. Model Validation Using T-37

The subjective validation was an iterative process, involving judgements about the model and its output by people who were knowledgeable about the T-37 and/or the T-46. These judgements were made based on reviews of flow (logic charts), input parameters, and assumptions. The sensitivity of the model outputs to changes in inputs was also reviewed.

FLEXIBILITY

Those inputs which are most likely to change or be the subject of sensitivity analysis are input using a data file.

There is no limit on the number of resource types, WUC's, or aircraft. Instructions for setting these limits are in Appendix F - Run Instructions.

The output header is formated in the INTLC subroutine and the output is defined in subroutine STATS. Both can easily be changed.

Because of the sortie scheduling subroutines the model is only applicable to an ATC operation. For use by other commands, the subroutines FLYING, FLYEND, FLYPM and the type of inputs (SCINFO array) would have to be changed. This model assumes only one plane is needed for each sortie; therefore, the subroutine SORTIE would have to be changed for formation flying.

OUTPUT

There are two output files - T46OUT LISTING and TSLAM LISTING.

The primary output file is called T46OUT LISTING. The following is an example of this file. This example is of 6 runs of 3000 hour duration. The MTBM varies on each run from 2.5 to 4.0.

The output file echos back some of the inputs (in order to identify the run), contains the primary parameters used in the IOT&E analysis (FMC rate, MDT/S, MMH/FH) and contains additional parameters which add enough detail to determine the reasonableness of the output. The output parameters which identify the output are the number of days simulated (DAYS), the mean time between maintenance (MTBM), the sum of all resources, except dummy resources (#RSCS), the not mission capable supply rate (NMCS). The parameters used in the IOT&E analysis are the fully mission capable rate (FMC), the mean downtime per sortie (MDT/S), and the maintenance manhours per flying hour (MMH/FH). The additional parameters are average number of items (planes or LRU's) waiting for maintenance (#WAIT), the average daily number of sorties missed due to maintenance (#MISS), the average cannibalization rate (CANN), the resource type causing the longest delay (LONG).

As discussed in the FLEXIBILITY section, the output can be easily modified to list the pertainent information needed for the analysis.

```
DAYS MTBM UTE #RSCS NMCS  FMC  MDT/S MMH/FH #WAIT #MISS CANN LONG
125  2.50  60   441  4.9  56.   5.9   4.51    77     7    5    4
125  2.70  60   441  4.8  72.   3.7   4.36    36     2    5    4
125  3.00  60   441  4.9  78.   3.0   4.02    25     1    5    4
125  3.20  60   441  5.0  78.   3.0   3.96    27     1    5    4
125  3.50  60   441  4.9  80.   2.5   3.75    20     1    5    3
125  4.00  60   441  4.8  83.   2.0   3.61    16     1    5    3
```

The other output file, TSLAM LISTING, is the standard SLAM output and contains statistics on all the files, activities, and resources. Although the output of primary interest is summarized in the T46OUT LISTING file, this file contains additional information which can be used to verify and validate the model.

APPENDIX A. LIST OF ACRONYMS

AFLC - Air Force Logistics Command

AFSC - Air Force Specialty Code

ATC - Air Training Command

CND - cannot duplicate

DOD - Department of Defense

FH - flying hours

FMC - Fully Mission Capable

IOT&E - Initial Operational Test and Evaluation

LRU - line replaceable unit

MDT/S - Mean Downtime Per Sortie

MMH/FH - Maintenance Manhours Per Flying Hour

MTBM - Mean Time Between Maintenance

NMCM - Not Mission Capable Maintenance

NMCS - Not Mission Capable Supply

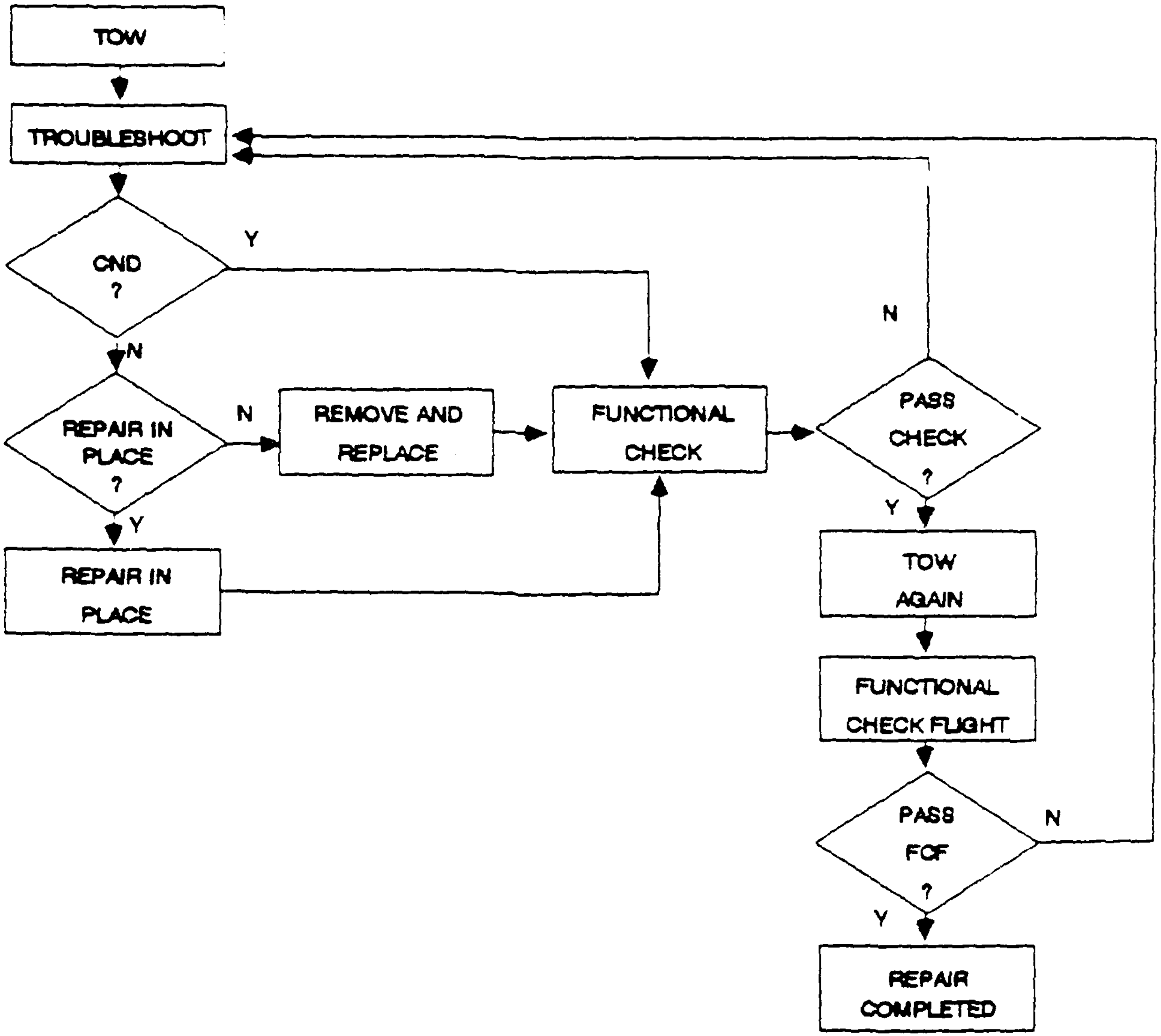
R/R - remove and replace

SLAM - Simulation Language for Alternative Modeling

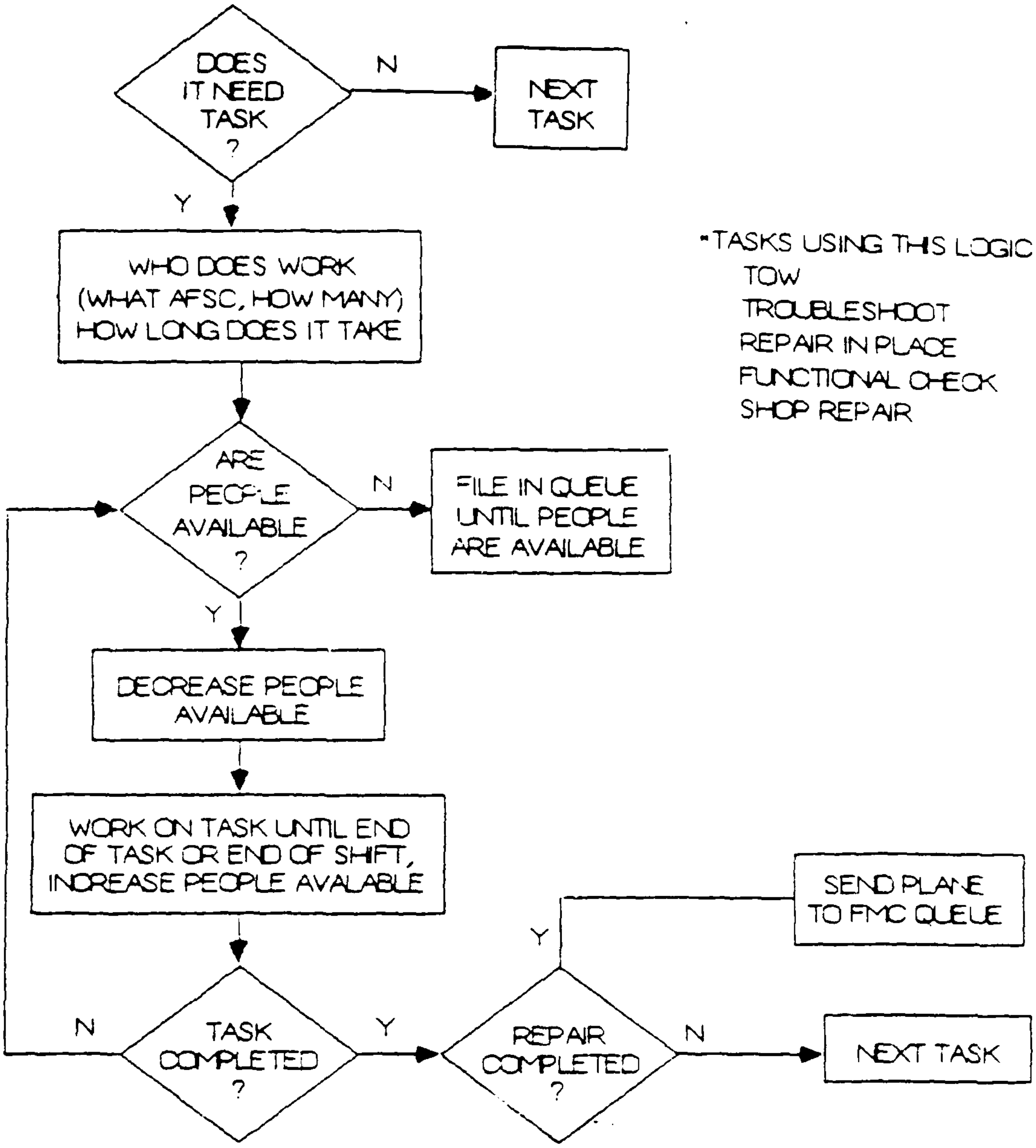
UPT - undergraduate pilot training

UTE - utilization rate (FH/aircraft/month)

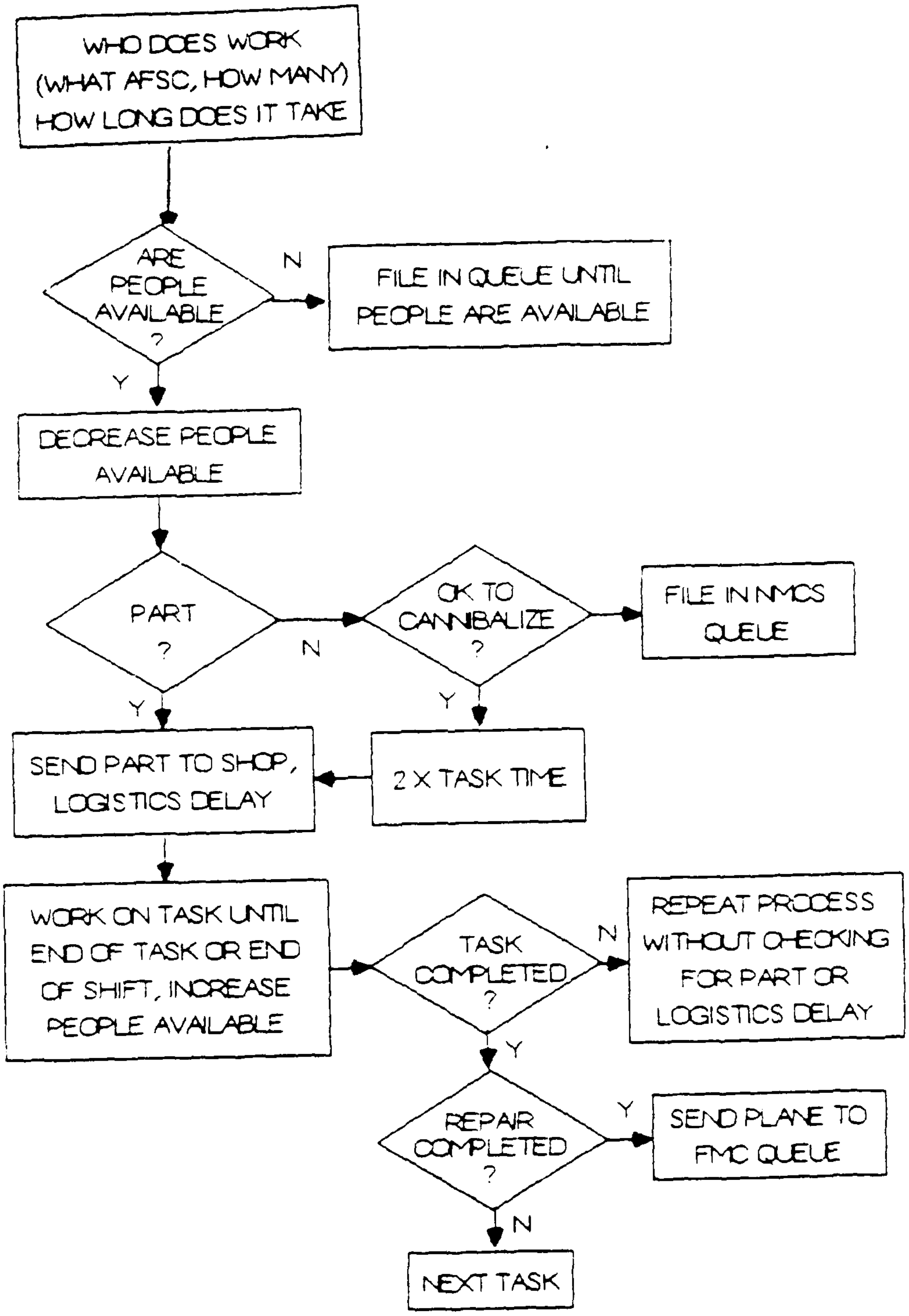
WUC - work unit code



Appendix B. Logic Charts
Summary Level Repair



Appendix B. Logic Charts
Repair Tasks


WHO DOES WORK
(WHAT AFSC, HOW MANY)
HOW LONG DOES IT TAKE
ARE PEOPLE AVAILABLE ?
N
FILE IN QUEUE UNTIL PEOPLE ARE AVAILABLE
Y
DECREASE PEOPLE AVAILABLE
PART ?
N
OK TO CANNIBALIZE ?
FILE IN NMCS QUEUE
Y
Y
SEND PART TO SHOP, LOGISTICS DELAY
2 X TASK TIME
WORK ON TASK UNTIL END OF TASK OR END OF SHIFT, INCREASE PEOPLE AVAILABLE
TASK COMPLETED ?
N
REPEAT PROCESS WITHOUT CHECKING FOR PART OR LOGISTICS DELAY
Y
REPAIR COMPLETED ?
Y
SEND PLANE TO FMC QUEUE
N
NEXT TASK

Appendix B. Logic Charts
Remove and Replace Task

APPENDIX C. COMPUTER CODE

```
***************************************************************
*             MAIN PROGRAM FOR T46A SLAM NETWORK
***************************************************************
*                           INTERIM      MATURE
*     AVAILABILITY            .37          .83
*     MTBM(C)                 .80FH       2.9FH
*     MMH/FH                15.3          4.75
*     MDT/S                 10.9HRS       3.0HRS
*     TURNAROUND TIME        TBD           14MIN
***************************************************************
      PROGRAM MAIN
      DIMENSION NSET(24000)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON QSET(24000)
      EQUIVALENCE(NSET(1),QSET(1))
      NNSET=24000
      NCRDR=5
      NPRNT=6
      NTAPE=7
      OPEN(7,STATUS='SCRATCH')
      CALL SLAM
      STOP
      END
************************************************************************
*                           INTLC                                      *
************************************************************************
      SUBROUTINE INTLC
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/T46AD1/JDATA(4)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
*** COMMON BLOCK GCOM1 IS USED TO SCHEDULE THE OUTPUT AT TIME TTFIN
      PARAMETER (MMXXV=100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(MMXXV),TTTS,TTTF
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
      COMMON/USTAT/AVGTRN,AVWAIT,DELTA,TOTTRN,TIMCLR,TFMC1
************************************************************************
      KDAY=1
      KMONTH=1
      NFLOWN=0
      NFLYDY=0
      NMISS=0
      NRSC=JDATA(1)
      NSPR=0
      NWEEK=1
      NWUC=JDATA(3)
      NWUC23=JDATA(4)
      TOTDT=0.0
      TOTFH=0.0
      TOTTRN=0.0
      XX(2)=3.0
      XX(6)=C.0
      PRINT*,'NUMBER OF PLANES=',PROGRM(1)
*** INITIALIZE # OF RESOURCES, COUNT TOTAL RSCS (EXCEPT DUMMY)
      NMEN=0
      DO 25 I=1,3
        DO 20 J=1,JDATA(1)
          NMEN=NMEN+MXINFO(I,J)
          JRSC(I,J)=0
          NWORK(I,J)=0
 20     CONTINUE
        NMEN=NMEN-MXINFO(I,JDATA(2))
 25   CONTINUE
***  CALCULATE # LOCAL AND CROSS COUNTRY (XC) SORTIES/YEAR
* NSORTS 1=ANNUAL LOCAL, 2=TODAY, 3=NOW, 4=LATER TODAY, 5=NIGHT,6=XC
      ZFH=PROGRM(1)*12*PROGRM(4)
      NSORTS(1)=.909*ZFH/1.3
      NSORTS(6)=(.09*ZFH/5.0)/52.0
*** CALC. OVERALL FAILURE RATE FROM INPUT FILE AND ADJUSTMENT
      ZFRATE=0.0
      DO 40 ITEM=1,NWUC
         ZFRATE=ZFRATE+1/CODES(ITEM,1)
 40   CONTINUE
      DELTA=XX(1)*ZFRATE
      PRINT*,'INPUT MTBM=',1/ZFRATE,'DELTA=',DELTA
*** FOR EACH WUC, SET NUMBER OF SPARES AVAILABLE, ASSIGN FAILURE
*** TIMES, CODE MANPOWER FOR DUMMY IF ONLY ONE AFSC NEEDED,
*** ADJUST MTBM'S BY DELTA (UNDONE AT TTFIN), CALC TOTAL MTBM,NMCS
      ZFRATE=0.0
      ZNMCS=0.0
      DO 55 ITEM=1,NWUC
         JSPARE(2,ITEM)=100
         CODES(ITEM,1)=CODES(ITEM,1)*DELTA
         FHFAIL(ITEM)=EXPON(CODES(ITEM,1),1)
         ZFRATE=ZFRATE+1/CODES(ITEM,1)
         ZNMCS=ZNMCS+CODES(ITEM,3)
         DO 50 J=1,11,2
           IF(JRSCTN(ITEM,J).EQ.0) THEN
             JRSCTN(ITEM,J)=JDATA(2)
             JRSCTN(ITEM,J+1)=1
           ENDIF
 50      CONTINUE
 55   CONTINUE
      XNMCSO=ZNMCS/NWUC
      PRINT*,'TOTAL MTBM=',1/ZFRATE,'NMCS%=',XNMCSO
*** CODE MANPOWER FOR DUMMY IF ONLY ONE TYPE USED FOR TASKS
      DO 60 J=1,12
         IF(TASKIN(J,3).EQ.0) THEN
            TASKIN(J,3)=JDATA(2)
            TASKIN(J,4)=1
         ENDIF
 60   CONTINUE
```

```
*** CREATE AIRCRAFT, ASSIGN #, INSPECTION TIME, FILE IN HOLD OR NMCS
      K=PROGRM(1)*(1.0-XNMCSO)
      NPLANE=PROGRM(1)
      XNWUC=NWUC
      DO 200 I=1,NPLANE
         ATRIB(1)=I
         ATRIB(2)=(I-1)*ACINFO(1)/PROGRM(1)
         ATRIB(7)=0.0
         ATRIB(17)=I*(-720.0/PROGRM(1))
         ATRIB(18)=I*(-200.0/PROGRM(1))
         IF (I.LE.K) THEN
            ATRIB(6)=0
            CALL FILEM(1,ATRIB)
         ELSE
            ATRIB(6)=1
            JPARTS(I,1)=UNFRM(1.0,XNWUC,1)
            CALL FILEM(10,ATRIB)
            ORDTIM=TRIAG(0.0,24.0,168.0,5)
            ATRIB(1)=999.0
            CALL SCHDL(10,ORDTIM,ATRIB)
         ENDIF
 200  CONTINUE
*** SCHEDULE FLYING, FLYEND, SHIFTS, STATS, STATSO ***
      ATRIB(1)=500.0
      CALL SCHDL(1,0.05,ATRIB)
      CALL SCHDL(3,23.5,ATRIB)
      CALL SCHDL(11,5.9,ATRIB)
      CALL SCHDL(12,6.0,ATRIB)
      CALL SCHDL(16,174.0,ATRIB)
      CALL SCHDL(17,TTFIN,ATRIB)
*** USE ATRIB1=999 FOR MONTH CHANGE
      ATRIB(1)=999.0
      CALL SCHDL(4,744.0,ATRIB)
*** SET UP OUTPUT FILE   ****
      WRITE(UNIT=20,FMT=300)NNRUN
 300  FORMAT(1X,'   BEGINNING RUN NUMBER',I2)
      IF(NNRUN.GT.1) RETURN
      WRITE(UNIT=9,FMT=360)
 360  FORMAT(1X,' DAYS MTBM UTE #RSCS NMCS  FMC  MDT/S MMH/FH #WAIT #MIS
     1S CANN LONG')
      RETURN
      END
**********************************************************************
*                              EVENT                                 *
**********************************************************************
      SUBROUTINE EVENT(I)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21),I
 1    CALL FLYING
      RETURN
 2    CALL FLYPM
      RETURN
 3    CALL FLYEND
      RETURN
 4    CALL CALNDR
```

```
      RETURN
 5    CALL SORTIE
      RETURN
 6    CALL CHECK1
      RETURN
 7    CALL CHECK2
      RETURN
 8    CALL TURN
      RETURN
 9    CALL SPPLY1
      RETURN
 10   CALL SPPLY2
      RETURN
 11   CALL STATS1
      RETURN
 12   CALL SHIFTS
      RETURN
 13   CALL MXSEIZ
      RETURN
 14   CALL MXFREE
      RETURN
 15   CALL MXQCK
      RETURN
 16   CALL STATSO
      RETURN
 17   CALL STATS
      RETURN
 18   CALL SPAREA
      RETURN
 19   CALL SPAREB
      RETURN
 20   CALL TASKNX
      RETURN
 21   CALL T38
      RETURN
      END
*********************************************************************
*                             ASSGNO
*********************************************************************
*     THIS SUBROUTINE ASSIGNS CODES AND CHANGES PROBABILITIES TO CODES.
*********************************************************************
      SUBROUTINE ASSGNO
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      ITEM=ATRIB(7)
      IF(ITEM.GT.NWUC) PRINT*,TNOW,'ASSGN,A1,ITEM=',ATRIB(1),ITEM
      DO 10 I=11,16
         J=I-7
         ATRIB(I)=CODES(ITEM,J)
         RAND=UNFRM(0.0,1.0,5)
         IF (RAND.LT.ATRIB(I)) ATRIB(I)=1
         IF (RAND.GE.ATRIB(I)) ATRIB(I)=0
 10   CONTINUE
      ATRIB(9)=0.0
      ATRIB(10)=0.0
      IF(ATRIB(16).EQ.1.0) ATRIB(25)=-1.0
      IF(ATRIB(16).NE.1.0) ATRIB(25)=0.0
      XNMCS=(FFAVG(10)+AAAVG(5))/PROGRM(1)
      RAND=UNFRM(0.0,1.0,5)
      IF(RAND.GT.1.4*CODES(ITEM,3)) ATRIB(10)=1
      IF(XNMCS.GT.XNMCSO*1.08) ATRIB(10)=1
      RETURN
      END
***********************************************************************
*                     CALNDR (EVENT 4)
***********************************************************************
*     CALNDR IS CALLED FROM INTLC WITH A CODE 999 TO INCREMENT THE
* THE MONTH COUNTER.
*     CALNDR INCREMENTS AND CHECKS THE DAY OF THE WEEK.  IF IT
* IS A WEEKDAY IT CALCULATES THE NUMBER OF SORTIES FOR THAT DAY
* CONSIDERING MONTHLY FLUCTUATIONS IN FLYING HOURS AND WEATHER
* CANCELLATIONS.  95% OF THE SORTIES ARE DAYTIME SORTIES.
* IT IS CALLED FROM FLYING AND RETURNS THE NUMBER OF MORNING SORTIES.
***********************************************************************
      SUBROUTINE CALNDR
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
*** IF CALLED TO INCREMENT MONTH COUNTER
      IF(ATRIB(1).EQ.999.0) THEN
         KMONTH=KMONTH+1
         CALL SCHDL(4,744.0,ATRIB)
         RETURN
      ENDIF
*** CALCULATE # OF SORTIES, ADD SORTIES MISSED YESTERDAY
      KDAY=KDAY+1
      IF (KDAY.LT.6) THEN
        CALL SCHDL(16,3.0,ATRIB)
        NFLYDY=NFLYDY+1
        ZWX1=100-SCINFO(3,KMONTH)
        ZWX2=100+SCINFO(3,KMONTH)
        ZWX=(UNFRM(ZWX1,ZWX2,1))/100.0
        ZSCH=SCINFO(1,KMONTH)/100.0
        NSORTS(2)=(NMISS-NMISS1)+NSORTS(1)*ZSCH/SCINFO(2,KMONTH)*ZWX
        NMISS1=NMISS
      ELSE
        NSORTS(2)=0
        IF(KDAY.EQ.7) THEN
          WRITE(UNIT=20,FMT=10)NWEEK,NNRUN
 10       FORMAT(1X,'      WEEK',I3,',  RUN # ',I2)
          NWEEK=NWEEK+1
          KDAY=0
        ENDIF
      ENDIF
      ZSORTS=.95*NSORTS(2)
      NSORTS(4)=ZSORTS/8
      NSORTS(3)=ZSORTS-7*NSORTS(4)
      NSORTS(5)=NSORTS(2)-ZSORTS
      RETURN
      END
************************************************************************
*               CHECK1, CHECK2 AND CHECK3
************************************************************************
*      CHECK1 CHECKS FOR PHASE INSPECTIONS AND FAILURES AFTER SORTIE.
*      CHECK2 CHECKS FOR A SECOND FAILURE AFTER A REPAIR AND CALCULATES
* THE MEAN DOWNTIME.  IT ALSO CHECKS FOR MISSING PARTS, DUE TO
* CANNIBALIZATION.  IF THE A/C IS NEEDED FOR A SORTIE, IT CHECKS
* IF THE REPAIR WAS ACCOMPLISHED WITHIN THE SORTIE CANCELLATION TIME.
*      EACH SUBROUTINE CALLS CHECK3, WHICH CHECKS THE FAILURE CLOCKS
* AND SENTS THE A/C TO THE REPAIR NETWORK, IF NECESSARY.
************************************************************************
      SUBROUTINE CHECK1
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      IF (NNQ(4).LT.1) PRINT*,TNOW,'ERROR, CHECK1'
      CALL RMOVE(1,4,ATRIB)
*** CHECK AIRCRAFT PHASE, ENGINE, AND PAINT
      IF (ATRIB(3).GT.ATRIB(2)) THEN
         ATRIB(2)=ATRIB(3)+ACINFO(1)
         ATRIB(7)=NWUC23
         PPAINT=.25*ACINFO(1)/(PROGRM(4)*12.0)
         RAND=UNFRM(0.0,1.0,1)
         IF(RAND.LT.PPAINT) THEN
           XX(6)=XX(6)+ACINFO(3)*4
           ATRIB(11)=1
         ELSE
           ATRIB(11)=0
         ENDIF
         RAND=UNFRM(0.0,1.0,1)
         IF(RAND.LT..5) ATRIB(25)=-3.0
         IF(RAND.GE..5) ATRIB(25)=-2.0
         ATRIB(4)=TNOW
         CALL ENTER(1,ATRIB)
         RAND=UNFRM(0,1.0,1)
         ENGPH=ACINFO(2)/ACINFO(1)
         IF(RAND.LE.ENGPH) THEN
           ATRIB(1)=300
           ATRIB(25)=-6
           CALL ENTER(1,ATRIB)
           CALL ENTER(1,ATRIB)
         ELSE
           CALL SPAREA
           CALL SPAREA
         ENDIF
         RETURN
      ENDIF
*** END OF CHECKING FOR PHASE
      KFROM=1
      CALL CHECK3 (KFAIL,KFROM)
      IF(KFAIL.EQ.1) RETURN
      XX(6)=XX(6)+TASKIN(8,5)
      ATRIB(4)=TNOW
      CALL SCHDL(8,TASKIN(8,5),ATRIB)
      RETURN
      END
************************************************************************
      SUBROUTINE CHECK2
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      KFROM=2
      CALL CHECK3 (KFAIL,KFROM)
      IF(KFAIL.EQ.1) RETURN
***   CHECK FOR CANNED PARTS
      IF(ATRIB(6).GT.0) THEN
          JTAIL=ATRIB(1)
          J=ATRIB(6)
          ITEM=JPARTS(JTAIL,J)
          ATRIB(7)=ITEM
          JPARTS(JTAIL,J)=0
          ATRIB(6)=ATRIB(6)-1.0
          CALL ASSGNO
          CALL ENTER(1,ATRIB)
          RETURN
      ENDIF
      DWNTM=TNOW-ATRIB(4)
      TOTDT=TOTDT+DWNTM
      XMDTPS=TOTDT/NFLOWN
      ATRIB(25)=0
      ATRIB(27)=0
      IF (ATRIB(8).EQ.4) THEN
         ATRIB(8)=2
         IF(DWNTM.LT..5) THEN
            XX(6)=XX(6)+.65
            ATRIB(3)=ATRIB(3)+1.3
            TOTFH=TOTFH+1.3
            NFLOWN=NFLOWN+1
            CALL SCHDL(6,1.3,ATRIB)
            CALL FILEM(4,ATRIB)
            RETURN
         ELSE
            NMISS=NMISS+1
            CALL FILEM(2,ATRIB)
            RETURN
         ENDIF
      ENDIF
      XX(6)=XX(6)+TASKIN(8,5)
      ATRIB(4)=TNOW
      CALL SCHDL(8,TASKIN(8,5),ATRIB)
      RETURN
      END
************************************************************************
      SUBROUTINE CHECK3 (KFAIL,KFROM)
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
** FOR PREFLIGHT, CHECK FH + 10 MINUTES
      IF (KFROM.EQ.0) CKTIME=TOTFH+0.167
      IF (KFROM.NE.0) CKTIME=TOTFH
      DO 10 ITEM=1,NWUC
      IF (FHFAIL(ITEM).LT.CKTIME) THEN
        IF(ITEM.GT.NWUC.OR.ITEM.LE.0) PRINT*,TNOW,'ERR, CHECK3'
        FHFAIL(ITEM)=TOTFH+EXPON(CODES(ITEM,1),1)
        IF (KFROM.EQ.0.AND.NNQ(3).LT.1) THEN
          ATRIB(8)=4
          KFLY=2
        ENDIF
        ATRIB(4)=TNOW
        ATRIB(7)=ITEM
        CALL ASSGNO
        CALL ENTER(1,ATRIB)
        IF (KFROM.EQ.0.AND.NNQ(3).GE.1) THEN
          CALL RMOVE(1,3,ATRIB)
          NSPR=NSPR+1
        ENDIF
        KFAIL=1
        RETURN
      ENDIF
 10   CONTINUE
      KFAIL=0
      RETURN
      END
************************************************************************
*        FLYING (EVENT 1), FLYPM (EVENT 5) AND FLYEND (EVENT 7)
************************************************************************
*     FLYING INITIATES THE DAILY FLYING SCHEDULE.  IT CALLS CALNDR,
* WHICH CALCULATES THE # OF SORTIES; PUTS A/C IN THE RDY AND SPR
* QUEUES; SCHEDULES NIGHT, CROSS COUNTRY AND 1/4 OF LOCAL SORTIES.
* IT SCHEDULES FLYPM THREE TIMES.
*     FLYPM SCHEDULES THE REMAINING LOCAL SORTIES.  (SCHEDULING THESE
* SORTIES SEPARATELY REDUCES THE MAXIMUM # OF EVENTS OF THE EVENT
* CALENDAR, USING LESS SPACE AND DECREASING RUN TIME.)
*     FLYEND THE A/C BACK TO THE HOLD QUEUES.
************************************************************************
      SUBROUTINE FLYING
      COMMON/SCOM1/A[illegible]0),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,N[illegible]NSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/[illegible]GRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/SYSTM/[illegible]INFO(3),TASKIN(12,5)
      COMMON/UCOM/F[illegible]AIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSP[illegible]NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      CALL SCHDL(1,2[illegible]0,ATRIB)
      CALL CALNDR
      IF(KDAY.EQ.0.OR.KDAY.GE.6) RETURN
      DO 20 IQ=2,3
        NTOQ=PROGRM(IQ)
        IF(KMONTH.GE.3.AND.KMONTH.LE.9) NTOQ=PROGRM(IQ)+2
        DO 10 I=1,NTOQ
          IF (NNQ(1).GT.0) THEN
```

```
         CALL RMOVE(1,1,ATRIB)
         XX(6)=XX(6)+TASKIN(7,5)
         ATRIB(8)=IQ
         CALL FILEM(IQ,ATRIB)
        ENDIF
 10    CONTINUE
 20   CONTINUE
      ATRIB(1)=1.3
      ATRIB(2)=1.3
      DO 30 I=1,NSORTS(3)
        TIMFLT=(SCINFO(4,KMONTH)/100.0)+(I*.05)
        CALL SCHDL(5,TIMFLT,ATRIB)
 30   CONTINUE
      IF (NSORTS(5).GT.0) THEN
       DO 40 J=1,NSORTS(5)
        TIMFLT=((SCINFO(5,KMONTH)/100.0)+(J*.05))
        CALL SCHDL(5,TIMFLT,ATRIB)
 40    CONTINUE
      ENDIF
      IF (KDAY.EQ.5) THEN
       ATRIB(1)=5.0
       ATRIB(2)=40.0
       DO 50 K=1,NSORTS(6)
        TIMFLT=16.5+K*.05
        CALL SCHDL(5,TIMFLT,ATRIB)
 50    CONTINUE
      ENDIF
      DO 60 ILATER=1,7
       ZHOUR=(SCINFO(5,KMONTH)-SCINFO(4,KMONTH))/100.0
       TIMSCH=(ZHOUR*ILATER*0.125)+SCINFO(4,KMONTH)/100.0
       CALL SCHDL(2,TIMSCH,ATRIB)
 60   CONTINUE
      RETURN
      END
***********************************************************************
      SUBROUTINE FLYPM
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      ZSORTS=.95*NSORTS(2)
      NSORTS(4)=ZSORTS/8
      ATRIB(1)=1.3
      ATRIB(2)=1.3
      DO 10 I=1,NSORTS(4)
        TIMFLT=I*.05
        CALL SCHDL(5,TIMFLT,ATRIB)
 10   CONTINUE
      RETURN
      END
***********************************************************************
      SUBROUTINE FLYEND
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
```

```
      CALL SCHDL(3,24.0,ATRIB)
      DO 50 IQ=2,3
        IF(NNQ(IQ).GT.0) THEN
          DO 25 I=1,NNQ(IQ)
            CALL RMOVE(1,IQ,ATRIB)
            IF (ATRIB(17)+720.0.LT.TNOW) THEN
               ATRIB(17)=TNOW
               XX(6)=XX(6)+TASKIN(10,5)
            ENDIF
            IF (ATRIB(18)+200.0.LT.ATRIB(3)) THEN
               ATRIB(18)=ATRIB(3)
               XX(6)=XX(6)+TASKIN(11,5)
            ENDIF
            XX(6)=XX(6)+TASKIN(9,5)
            ATRIB(8)=1.0
            CALL FILEM(1,ATRIB)
 25       CONTINUE
        ENDIF
 50   CONTINUE
      RETURN
      END
************************************************************************
*      MXSEIZ (EVENT 13), MXFREE (EVENT 14), AND MXQCK (EVENT 15)
**********************************************************************
*  MXSEIZ SEIZES THE RESOURCES, IF AVAILABLE; IF NOT, FILES IN QUEUES.
*  MXFREE IS CALLED AFTER THE PHASE OF THE REPAIR IS FINISHED AND
*FREES THE RESOURCES AND CALCULATES THE MANHOURS.
*  MXQCK CHECKS IF RESOURCES ARE NOW AVAILABLE (CALLED AT THE START OF
*EACH SHIFT AND AFTER MXFREE FREES RESOURCES).
**********************************************************************
      SUBROUTINE MXSEIZ
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
      INTEGER ITIMES(4),IRSCTN(4)
*** CODES FOR READING THE INPUT FILE
      DATA ITIMES/1,4,7,10/
      DATA IRSCTN/1,5,5,9/
      IF(ATRIB(1).GE.380.) GO TO 10
      ITEM=ATRIB(7)
      IF(ITEM.GT.65)PRINT*,TNOW,'MXSEIZ,A1,ITEM=',ATRIB(1),ITEM
*** DETERMINE WHAT RESOURCES AND TIMES ARE NEEDED
      IF(ATRIB(25).LT.0.OR.ATRIB(25).EQ.4) THEN
         KTASK=ATRIB(25)+7
         IF(ATRIB(25).EQ.4) KTASK=12
         ATRIB(21)=TASKIN(KTASK,1)
         ATRIB(22)=TASKIN(KTASK,2)
         ATRIB(23)=TASKIN(KTASK,3)
         ATRIB(24)=TASKIN(KTASK,4)
         IF(ATRIB(26).EQ.0) ATRIB(26)=TASKIN(KTASK,5)
      ELSE IF (ATRIB(25).EQ.6) THEN
         ATRIB(21)=SHOP(ITEM,1)
         ATRIB(22)=SHOP(ITEM,2)
         ATRIB(23)=12
         ATRIB(24)=1
         IF(ATRIB(26).EQ.0) THEN
           ZA=SHOP(ITEM,3)
           ZB=SHOP(ITEM,4)
           ZC=SHOP(ITEM,5)
           IF(ZC.LE.ZB.OR.ZB.LE.ZA) PRINT*,TNOW,'ERR, MXSEIZ',ITEM
           ATRIB(26)=TRIAG(ZA,ZB,ZC,1)/60.0
        ENDIF
      ELSE
        I=ATRIB(25)+1
        ATRIB(21)=JRSCTN(ITEM,IRSCTN(I))
        ATRIB(22)=JRSCTN(ITEM,IRSCTN(I)+1)
        ATRIB(23)=JRSCTN(ITEM,IRSCTN(I)+2)
        ATRIB(24)=JRSCTN(ITEM,IRSCTN(I)+3)
*** ASSIGN TASK TIME, UNLESS FINISHING TASK FROM PREVIOUS SHIFT
        IF(ATRIB(26).EQ.0) THEN
          ZA=TIMES(ITEM,ITIMES(I))
          ZB=TIMES(ITEM,ITIMES(I)+1)
          ZC=TIMES(ITEM,ITIMES(I)+2)
          IF(ZB.GT.0.AND.ZC.GT.0) THEN
            IF(ZC.LE.ZB.OR.ZB.LE.ZA) PRINT*,TNOW,'ERRA,MXSEIZ',ITEM
            ATRIB(26)=TRIAG(ZA,ZB,ZC,5)/60.0
          ELSE
            ATRIB(26)=0.0
          ENDIF
        ENDIF
      ENDIF
*** END OF DETERMINING RESOURCES AND TIMES
 10   CONTINUE
      ISHIFT=XX(2)
      IR1=ATRIB(21)
      NR1=ATRIB(22)
      IR2=ATRIB(23)
      NR2=ATRIB(24)
      IF (JRSC(ISHIFT,IR1).GE.NR1.AND.JRSC(ISHIFT,IR2).GE.NR2) THEN
        IF (ATRIB(25).EQ.2.AND.ATRIB(9).NE.1) THEN
          RAND=UNFRM(0.0,1.0,1)
          IF (RAND.LT..83) ATRIB(5)=0.5+EXPON(0.3,1)
          IF (RAND.GE..83) ATRIB(5)=1.0+EXPON(2.0,1)
          IF (ATRIB(5)+TNOW.GE.XX(3)) ATRIB(5)=XX(3)-TNOW
          ENDTSK=TNOW+ATRIB(5)+ATRIB(26)
        ELSE
          ATRIB(5)=0.0
          ENDTSK=TNOW+ATRIB(26)
        ENDIF
        IF (XX(3).GT.ENDTSK) THEN
          ATRIB(27)=0
        ELSE
          ATRIB(26)=XX(3)-TNOW-ATRIB(5)-.001
          ATRIB(27)=ENDTSK-XX(3)
        ENDIF
        JRSC(ISHIFT,IR1)=JRSC(ISHIFT,IR1)-NR1
        JRSC(ISHIFT,IR2)=JRSC(ISHIFT,IR2)-NR2
      ELSE
        IF(JRSC(ISHIFT,IR1).LT.NR1) IQ=IR1+10+(ISHIFT-1)*NRSC
        IF(JRSC(ISHIFT,IR2).LT.NR2) IQ=IR2+10+(ISHIFT-1)*NRSC
        CALL FILEM(IQ,ATRIB)
        ATRIB(8)=99
      ENDIF
      RETURN
      END
************************************************************************
      SUBROUTINE MXFREE
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
      INTEGER IR(2),NR(2)
      ISHIFT=XX(2)
      IF (ATRIB(21).GT.25.OR.ATRIB(23).GT.25) PRINT*,TNOW,'ERR,MXFREE'
      IR(1)=ATRIB(21)
      NR(1)=ATRIB(22)
      IR(2)=ATRIB(23)
      NR(2)=ATRIB(24)
* DONT COUNT MANHOURS FOR DUMMY OR SE OR T38
      DO 10 J=1,2
        JRSC(ISHIFT,IR(J))=JRSC(ISHIFT,IR(J))+NR(J)
        IF(IR(J).LT.12.AND.ATRIB(1).LT.380.) THEN
           XMMH=ATRIB(26)*NR(J)
           IF(IR(J).EQ.9)XMMH=4*XMMH
        ELSE
           XMMH=0
        ENDIF
        XX(6)=XX(6)+XMMH
 10   CONTINUE
*** RESET A36, A37
      ATRIB(26)=ATRIB(27)
      ATRIB(27)=0
*** IF FINISHING R&R ON LATER SHIFT, CODE FOR NO DELAY OR PART NEEDED
      IF(ATRIB(26).NE.0.AND.ATRIB(25).EQ.2.0) ATRIB(9)=1.0
*** SCHEDULE NMCM TO CHECK IF ANY WAITING FOR RESOURCES JUST RELEASED
      CALL SCHDL(15,.06,ATRIB)
      RETURN
      END
*********************************************************************
      SUBROUTINE MXQCK
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
      ISHIFT=XX(2)
      ISTART=11+(ISHIFT-1)*NRSC
      ISTOP=ISTART-1+NRSC
      DO 20 IQ=ISTART,ISTOP
      IF (NNQ(IQ).GT.0) THEN
        DO 10 IPLANE=1,NNQ(IQ)
          CALL COPY(IPLANE,IQ,ATRIB)
          IR1=ATRIB(21)
          NR1=ATRIB(22)
          IR2=ATRIB(23)
          NR2=ATRIB(24)
          IF(IR1.GT.13.OR.IR2.GT.13) PRINT*,'ERR,MXQCK'
          IF(JRSC(ISHIFT,IR1).GE.NR1.AND.JRSC(ISHIFT,IR2).GE.NR2)THEN
            CALL RMOVE(IPLANE,IQ,ATRIB)
            CALL ENTER(1,ATRIB)
            CALL SCHDL(15,.01,ATRIB)
            RETURN
          ENDIF
 10     CONTINUE
      ENDIF
 20   CONTINUE
      RETURN
      END
************************************************************************
*                SHIFTS (EVENT 12) AND SHIFTQ
************************************************************************
*    SHIFTS ZEROS THE MANPOWER FOR THE CURRENT SHIFT, MOVES THE A/C
* WAITING QUEUES FROM ONE SET TO ANOTHER, CHANGES TO THE
* NEXT SHIFT, CALLS SHIFTQ, AND SETS MANPOWER FOR CURRENT SHIFT
* TO FULL STRENGTH. THE MANPOWER IS THEN ADJUSTED TO ACCOUNT FOR
* TRAINING, LEAVE, ETC.
************************************************************************
      SUBROUTINE SHIFTS
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
      ISHIFT=XX(2)
      DO 10 J=1,NRSC
        JRSC(ISHIFT,J)=0
 10   CONTINUE
***  AT START OF SHIFT, SCHDL MXQCK AND RESCHEDULE SHIFTS ***
*** MOVE A/C FROM ONE SET OF QUEUES TO ANOTHER FOR NEW SHIFT
      ISTART=11+(ISHIFT-1)*NRSC
      ISTOP=ISTART-1+NRSC
      DO 15 IQ=ISTART,ISTOP
        IF(NNQ(IQ).GT.0) THEN
          DO 12 J=1,NNQ(IQ)
            CALL RMOVE(1,IQ,ATRIB)
            IF(ISHIFT.EQ.3)IQNEW=IQ-2*NRSC
            IF(ISHIFT.NE.3)IQNEW=IQ+NRSC
            CALL FILEM(IQNEW,ATRIB)
 12       CONTINUE
        ENDIF
 15   CONTINUE
*** CHANGE SHIFT, DETERMINE # RESOURCES FOR THE SHIFT
      XX(2)=XX(2)+1
      IF(XX(2).EQ.4) XX(2)=1
      ISHIFT=XX(2)
      DO 20 J=1,13
        JRSC(ISHIFT,J)=MXINFO(ISHIFT,J)
 20   CONTINUE
      DO 30 IR=1,11
        IF (JRSC(ISHIFT,IR).GT.0) THEN
          DO 25 J=1,JRSC(ISHIFT,IR)
            RAND=100.0*UNFRM(0.0,1.0,5)
            IF (RAND.GT.MXINFO(4,IR)) THEN
              JRSC(ISHIFT,IR)=JRSC(ISHIFT,IR)-1
            ENDIF
 25       CONTINUE
```

```
        ENDIF
        NWORK(ISHIFT,IR)=JRSC(ISHIFT,IR)
 30   CONTINUE
***  AT START OF SHIFT, SCHDL MXQCK AND RESCHEDULE SHIFTS ***
      CALL SCHDL(15,.02,ATRIB)
      IF (ISHIFT.NE.3) CALL SCHDL(21,.01,ATRIB)
      IF(ISHIFT.EQ.1) SFTCHG=SFTLEN(1)
      IF(ISHIFT.EQ.2) SFTCHG=SFTLEN(2)
      IF(ISHIFT.EQ.3.AND.KDAY.NE.5) SFTCHG=SFTLEN(3)
      IF(ISHIFT.EQ.3.AND.KDAY.EQ.5) SFTCHG=SFTLEN(3)+SFTLEN(4)
      CALL SCHDL(12,SFTCHG,ATRIB)
** CHECK # USED AT SHIFT END
      XX(3)=TNOW+SFTCHG
      RETURN
      END
**********************************************************************
*                    SORTIE (EVENT 2)
**********************************************************************
*    SORTIE INITIATES EACH SORTIE BY LOCATING AN AIRCRAFT, CALLING
* CHECK3 (WHICH CHECKS FOR FAILURES), INCREMENTING THE FH, SORTIES,
* AND MMH; SENDING IT TO THE FLY NETWORK AND SCHEDULING SORTIE END(EV4).
**********************************************************************
      SUBROUTINE SORTIE
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
*** SORTIE LENGTH AND DURATION ARE THE ATRIBS FROM THE EVENT CALENDAR,
* NOT FROM THE ENTITY
      SORTA=ATRIB(1)
      SORTB=ATRIB(2)
      KFLY=0
      IF (NNQ(2).GT.0) THEN
         CALL RMOVE(1,2,ATRIB)
      ELSE IF (NNQ(3).GT.0) THEN
         NSPR=NSPR+1
         CALL RMOVE(1,3,ATRIB)
      ELSE
         KFLY=1
         RETURN
      ENDIF
      KFROM=0
      CALL CHECK3 (KFAIL,KFROM)
      IF (KFLY.EQ.0) THEN
*** LAUNCH AND RECOVERY (SHOULD NOT CHANGE WITH AIRCRAFT)
          XX(6)=XX(6)+0.65
          ATRIB(3)=ATRIB(3)+SORTA
          TOTFH=TOTFH+SORTA
          NFLOWN=NFLOWN+1
          CALL SCHDL(6,SORTB,ATRIB)
          CALL FILEM(4,ATRIB)
      ELSE IF (KFLY.EQ.1) THEN
          NMISS=NMISS+1
      ENDIF
      RETURN
      END
```

```
************************************************************************
*       SPAREA (EVENT 18) AND SPAREB (EVENT 20)
************************************************************************
*   SPAREA INCREMENTS SPARES AVAILABLE AND DECREMENTS DEMAND. IT IS
* CALLED FOLLOWING REPAIR OF A SPARE.
************************************************************************
      SUBROUTINE SPAREA
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      ITEM=ATRIB(7)
      JSPARE(2,ITEM)=JSPARE(2,ITEM)+1
      JSPARE(3,ITEM)=JSPARE(3,ITEM)-1
      RETURN
      END
************************************************************************
      SUBROUTINE SPAREB
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      NCALLS=NCALLS+1
      ITEM=ATRIB(7)
      JSPARE(3,ITEM)=JSPARE(3,ITEM)+1
      IF(JSPARE(4,ITEM).LT.JSPARE(3,ITEM))JSPARE(4,ITEM)=JSPARE(3,ITEM)
      IF(JSPARE(2,ITEM).GT.0) THEN
        JSPARE(2,ITEM)=JSPARE(2,ITEM)-1
      ELSE
        PRINT*,TNOW,'NEED SOMETHING IN SPAREB'
*       CALL FILEM(10,ATRIB)
*       ATRIB(8)=99
      ENDIF
      RETURN
      END
************************************************************************
*       SPPLY1 (EVENT 9), SPPLY2 (EVENT 10), AND SPPLY3 (EVENT 11)
************************************************************************
*    SPPLY1 FINDS AN AIRCRAFT FROM WHICH TO CANNIBALIZE.
* IT CHECKS THE # OF ITEMS MISSING, AND IF THE ITEM
* NEEDED IS ON THE A/C.  XX(4) IS THE NUMBER OF THE ITEM NEEDED.
*    SPPLY2 (IF CALLED FROM THE NETWORK) INCREMENTS THE NUMBER OF
* PARTS MISSING, ASSIGNS A CODE FOR THE PART MISSING, FILES THE AIRCRAFT
* IN THE NMCS QUEUE, SCHEDULES THE TIME THE PART WILL COME IN, AND
* SENDS A DUMMY ENTITY BACK TO THE NETWORK TO RELEASE THE MANPOWER.
*    SPPLY2 (IF SCHEDULED IN SPPLY2) REMOVES PLANE FROM NMCS QUEUE
* AND SENDS IT TO THE REPAIR NETWORK.
************************************************************************
      SUBROUTINE SPPLY1
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      XX(5)=0
      ITEM=ATRIB(7)
      IF(CODES(ITEM,2).GT.0) THEN
       IF (NNQ(10).GE.1) THEN
        DO 20 IPLANE=1,NNQ(10)
           CALL COPY(IPLANE,10,ATRIB)
           IF (ATRIB(6).LE.9) THEN
             JTAIL=ATRIB(1)
             DO 10 J=1,10
               IF (JPARTS(JTAIL,J).EQ.XX(4)) THEN
                 GO TO 20
               ENDIF
 10          CONTINUE
             NKBALL=NKBALL+1
             XX(5)=1
             CALL RMOVE(IPLANE,10,ATRIB)
             ATRIB(6)=ATRIB(6)+1
             J=ATRIB(6)
             JPARTS(JTAIL,J)=XX(4)
             CALL FILEM(10,ATRIB)
             XX(4)=0
             RETURN
          ENDIF
 20   CONTINUE
      ENDIF
      ENDIF
      RETURN
      END
**************************************************************************
      SUBROUTINE SPPLY2
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCSO
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
* IF CALLED FROM SPPLY2 BECAUSE PART IS IN (USE FIFO)
      IF(ATRIB(1).EQ.999.0) THEN
        IF (NNQ(10).EQ.0) PRINT*,TNOW,'ERROR IN SPPLY3'
        CALL RMOVE(1,10,ATRIB)
        JTAIL=ATRIB(1)
        J=ATRIB(6)
        ATRIB(7)=JPARTS(JTAIL,J)
        ATRIB(6)=ATRIB(6)-1
        ITEM=ATRIB(7)
        CALL ASSGNO
        CALL ENTER(1,ATRIB)
      ELSE
*** IF CALLED FROM NETWORK TO ORDER PART
        IF(ATRIB(8).NE.4) ATRIB(8)=0.0
        ATRIB(6)=ATRIB(6)+1
        JTAIL=ATRIB(1)
        J=ATRIB(6)
        JPARTS(JTAIL,J)=ATRIB(7)
        CALL FILEM(10,ATRIB)
        XNMCS=(FFAVG(10)+AAAVG(5))/PROGRM(1)
        ORDTIM=TRIAG(96.0,120.0,168.0,5)
        IF(XNMCS.GT.XNMCSO*1.05) ORDTIM=TRIAG(24.0,36.0,54.0,5)
        IF(XNMCS.LE.XNMCSO*0.95) ORDTIM=TRIAG(240.0,360.0,500.0,5)
        ATRIB(1)=999.0
        CALL SCHDL(10,ORDTIM,ATRIB)
        ATRIB(8)=99
      ENDIF
      RETURN
      END
************************************************************************
*                     STATS0 (EVENT 16)
************************************************************************
      SUBROUTINE STATS0
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/USTAT/AVGTRN,AVWAIT,DELTA,TOTTRN,TIMCLR,TFMC1
      COMMON/USFTS/JRSC(3,13),NRSC,NWORK(3,13)
*** CHECK FMC
      NTURN=0
      IF(NNQ(50).GT.0) THEN
        DO 30 IPLANE=1,NNQ(50)
          CALL COPY(IPLANE,50,ATRIB)
          IF(ATRIB(28).EQ.8.0) NTURN=NTURN+1
 30     CONTINUE
      ENDIF
      TFMCP=(NNQ(1)+NNQ(2)+NNQ(3)+NNQ(4)+NTURN)/PROGRM(1)
      TFMC=TFMC+TFMCP
      NOBSV=NOBSV+1
      TFMC1=100.0*TFMC/NOBSV
      RETURN
      END
************************************************************************
*                     STATS1 (EVENT 12)
************************************************************************
*  THIS SUBROUTINE CHECKS THE NUMBER OF AIRCRAFT WAITING TO BE PRE-
* FLIGHTED AT SUNRISE (BETTER THAN FFAVG(47) SINCE IT DOESN'T MATTER
* HOW LONG THEY WAIT, JUST SO THEY'RE DONE BY SUNRISE.)
************************************************************************
      SUBROUTINE STATS1
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/USTAT/AVGTRN,AVWAIT,DELTA,TOTTRN,TIMCLR,TFMC1
      IF(TNOW.LT.24.0) TTWAIT=0.0
      IF(TNOW.LT.24.0) NOBSV=0
      CALL SCHDL(11,24.0,ATRIB)
      TTWAIT=TTWAIT+NNQ(47)
      NOBSV=NOBSV+1
      AVWAIT=TTWAIT/NOBSV
      RETURN
      END
************************************************************************
*                      STATS (EVENT 17)
************************************************************************
*     STATS CALCULATES AND/OR PRINTS OUT FMC, MMH/FH, MDT/S,
* MAX. AND MIN. QUEUE LENGTHS FOR MANPOWER, AND AVG. SORTIES MISSED.
* IT IS SCHEDULED INITIALLY IN INTLC AT TIME TTFIN (END OF RUN).
************************************************************************
      SUBROUTINE STATS
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(1C0),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
      COMMON/UCOM/FHFAIL(65),KDAY,KFLY,KMONTH,NFLOWN,NFLYDY,NMEN,NMISS
     1,NSORTS(6),NSPR,NWEEK,NWUC,NWUC23,TOTDT,TOTFH,XMDTPS,XNMCS0
      COMMON/USPR/JPARTS(150,10),JSPARE(4,65),NKBALL,NCALLS
      COMMON/USTAT/AVGTRN,AVWAIT,DELTA,TOTTRN,TIMCLR,TFMC1
      REAL MMHPFH,MAXQL,MINQL
*** PRINT SPARES DEMAND
      DO 10 ITEM=1,NWUC
        PRINT*,'ITEM=',ITEM,'MAX. DEMAND=',JSPARE(4,ITEM)
 10   CONTINUE
***      CALCULATE OUTPUT PARAMETERS       ***
      NEND=TNOW/24.0
*** UNDO MTBM ADJUSTMENT
      ZFRATE=0.0
      DO 5 ITEM=1,NWUC
        ZFRATE=ZFRATE+1/CODES(ITEM,1)
        CODES(ITEM,1)=CODES(ITEM,1)/DELTA
  5   CONTINUE
      XMTBM=1/ZFRATE
      JUTE=PROGRM(4)
      TOTFMC=FFAVG(1)+FFAVG(2)+FFAVG(3)+FFAVG(4)+AVGTRN
      AVGFMC=100.0*TOTFMC/PROGRM(1)
      MMHPFH=XX(6)/TOTFH
      XWAIT=0.0
      DO 15 IQ=11,49
        XWAIT=XWAIT+FFAVG(IQ)
 15   CONTINUE
      JWAIT=XWAIT
      JCANN=100*NKBALL/NCALLS
      JMISS=100*NMISS/NFLOWN
      JSPR=NSPR/NFLYDY
      XNMCS1=100.0*((FFAVG(10)+AAAVG(5))/PROGRM(1))
      XLONG=0.0
      DO 25 IQ=11,23
        ZLONG=FFAVG(IQ)+FFAVG(IQ+13)+FFAVG(IQ+26)
        IF(ZLONG.GT.XLONG) THEN
          XLONG=ZLONG
          JLONG=IQ-10
        ENDIF
 25   CONTINUE
      WRITE(UNIT=9,FMT=310)NEND,XMTBM,JUTE,NMEN,XNMCS1,AVGFMC
     1,XMDTPS,MMHPFH,JWAIT,JMISS,JCANN,JLONG
 310  FORMAT(2X,I3,2X,F4.2,2X,I2,2X,I4,2X,F3.1,2X,F3.0,3X,F3.1,3X
     1,F4.2,4X,I2,4X,I2,3X,I2,3X,I2)
      RETURN
      END
************************************************************************
      SUBROUTINE TASKNX
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      ATRIB(25)=ATRIB(25)+1.0
***** DONT PUT FUNCTIONAL CK IN NESTED LOOP
*** AFTER ENGINE WORK OR SHOP WORK, INCREMENT SPARES
      IF(ATRIB(25).EQ.-3.OR.ATRIB(25).EQ.7) THEN
        CALL SCHDL(18,.01,ATRIB)
        ATRIB(8)=99.0
*** AFTER MAJOR OR MINOR PHASE, GET 2 ENGINES, SENT TO PAINT OR HOLD
      ELSE IF (ATRIB(25).EQ.-2.OR.ATRIB(25).EQ.-1) THEN
```

```
        NPHASE=NPHASE+1
        CALL SPAREB
        CALL SPAREB
        IF(ATRIB(11).EQ.1) CALL FILEM(9,ATRIB)
        IF(ATRIB(11).NE.1) CALL SCHDL(7,.001,ATRIB)
        ATRIB(8)=99.0
*** AFTER TROUBLESHOOT, IF CND, SKIP TO FUNCTIONAL CHECK
      ELSE IF (ATRIB(25).EQ.1.0.AND.ATRIB(15).EQ.1.0) THEN
        ATRIB(25)=ATRIB(25)+2.0
*** AFTER TROUBLESHOOT, A11 = 0 MEANS NO EQUIPMENT REPAIR, MUST R2
      ELSE IF (ATRIB(25).EQ.1.0.AND.ATRIB(11).EQ.0) THEN
        ATRIB(25)=ATRIB(25)+1.0
*** GO FROM ONAC TO FUNCTIONAL CHECK
      ELSE IF(ATRIB(25).EQ.2) THEN
        ATRIB(25)=ATRIB(25)+1.0
      ENDIF
*** AFTER REPAIR, IF NO FUNCTIONAL CHECK, GO TO TOW
      IF (ATRIB(25).EQ.3.AND.ATRIB(12).EQ.0) THEN
        ATRIB(25)=ATRIB(25)+1.0
      ENDIF
*** REPAIR COMPLETED
      IF(ATRIB(25).EQ.4.AND.ATRIB(16).EQ.0) ATRIB(25)=ATRIB(25)+1.0
      IF (ATRIB(25).EQ.5) THEN
*** FAILED FUNCTIONAL CHECK, START OVER
         IF(ATRIB(15).EQ.1.0) THEN
           ATRIB(15)=0
           ATRIB(25)=0
           CALL ENTER(1,ATRIB)
*** FINISHED REPAIR, CALL CHECK
         ELSE IF(ATRIB(13).EQ.0) THEN
           CALL SCHDL(7,.001,ATRIB)
*** NEED FCF
         ELSE IF(ATRIB(13).GT.0) THEN
           ATRIB(3)=ATRIB(3)+0.8
           RAND=UNFRM(0.0,1.0,1)
           IF (RAND.LT..91) THEN
             CALL SCHDL(7,1.3,ATRIB)
           ELSE
             ATRIB(25)=0.0
             CALL ENTER(4,ATRIB)
           ENDIF
         ENDIF
         ATRIB(8)=99.0
      ENDIF
      RETURN
      END
***********************************************************************
      SUBROUTINE TURN
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/USTAT/AVGTRN,AVWAIT,DELTA,TOTTRN,TIMCLR,TFMC1
      TOTTRN=TOTTRN+TNOW-ATRIB(4)
      AVGTRN=TOTTRN/TNOW
      IF (ATRIB(8).EQ.3) THEN
         CALL FILEM(3,ATRIB)
      ELSE
         CALL FILEM(2,ATRIB)
      ENDIF
      RETURN
      END
************************************************************************
      SUBROUTINE T38
      COMMON/SCOM1/ATRIB(100),DD(100),DDL(100),DTNOW,II,MFA,MSTOP,NCLNR
     1,NCRDR,NPRNT,NNRUN,NNSET,NTAPE,SS(100),SSL(100),TNEXT,TNOW,XX(100)
      COMMON/T46ADI/JDATA(4)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      DO 10 I=1,JDATA(1)
        ATRIB(1)=380.0+I
        ATRIB(21)=I
        ATRIB(22)=2
        ATRIB(23)=JDATA(2)
        ATRIB(24)=1
        ATRIB(25)=38
*** 520 SHIFTS PER YEAR X 2 PEOPLE X 4 ITEMS
        ATRIB(26)=MXINFO(5,I)/4160.0
        IF(ATRIB(26).GT.0) THEN
          DO 5 I1=1,4
            CALL ENTER(1,ATRIB)
  5       CONTINUE
        ENDIF
 10   CONTINUE
      RETURN
      END
```

```
GEN,GJERSTAD,T46,08/05/86,1,N,N,Y,N,Y,72;
LIMITS,49,27,450;  MFILS,MATRS,MENTS
SEEDS,0(1)/YES,0(5)/YES;
INTLC,XX(1)=2.9,XX(5)=99.0; MTBM, NO KBALL
NETWORK;
;***************************************************************
;VARIABLES
;  1  MTBM, TOTAL CORRECTIVE        4  CODE FOR PART NEEDED
;  2  SHIFT NUMBER                  5  CODE FOR KBALL(1=YES,0=NO)
;  3  END OF CURRENT SHIFT          6  CUMULATIVE MAINT. MANHOURS
;ATTRIBUTES
;  1  A/C#, LRU=200, ENGINE=300   13  FUNCTIONAL CHECK FLIGHT
;  2  INSPECTION TIME             14  CND
;  3  TOTAL FLYING HOURS          15  REJECT AFTER FUNCTIONAL CHECK
;  4  TIME FLOWN, OR TURN STARTED 16  TOWED FOR REPAIR
;  5  LOGISTICS DELAY TIME        17  DAY OF LAST SCHD MX
;  6  # ITEMS MISSING             18  FH SINCE LAST SCHD MX
;  7  ITEM TO BE REPAIRED      19,20  NOT USED
;  8  STATUS (2=SCHDL,3=SPARE,4=NEED SORTIE,99=DUMMY)
;  9  R2 CODE (1=HAVE PART)    21-24  RESOURCE CODES
; 10  SPARE PART AVAILABLE        25  TASK CODE
; 11  ON AIRCRAFT REPAIR          26  TASK TIME THIS SHIFT
; 12  FUNCTIONAL TEST             27  TASK TIME NEXT SHIFT
;ATTRIBUTES FOR CALENDAR EVENTS
;  EVENT  2  A1=SORTIE LENGTH  A2=COMPLETION TIME
;FILES:                        DAY EVEN NITE
;  1  A/C NOT SCHD TO FLY       11  24   37    INSTRUMENTS
;  2  A/C READY TO FLY          12  25   38    COMNAV
;  3  SPARE A/C, PREFLIGHTED    13  26   39    ELECTRICS
;  4  HELD HERE DURING SORTIE   14  27   40    ECS
;  5  NOT USED                  15  28   41    EGRESS
;  6  NOT USED                  16  29   42    FUEL
;  7  NOT USED                  17  30   43    HYDRAULICS
;  8  NOT USED                  18  31   44    ENGINES
;  9  PHASE INSPECTION          19  32   45    PHASE DOCK
; 10  A/C WAITING PARTS         20  33   46    SHT MTL, MACH
;                               21  34   47    APG
;                               22  35   48    DUMMY
;                               23  36   49    AUTOECS
;**************************************************************
;QUEUES
HOLD  QUEUE(1);
;PAINT NETWORK
PNT   QUEUE(9);    PHASE
      ACT/21,96.0,,HOLD; PAINT
;REPAIR NETWORK
;   THIS NETWORK IS USED FOR SHOP REPAIR, TOWING, TROUBLESHOOT,
;REPAIR ON AIRCRAFT, REMOVE AND REPLACE, FUNCTION CHECK, AIRFRAME
;PHASE(MAJOR, MINOR, PAINT), ENGINE WORK (TEARDOWN/BUILDUP,
;AUTOECS AND TEST CELL).
;   THE PEOPLE AND SE NEEDED AND THE TASK TIME IS DETERMINED
;IN THE FORTRAN, BASED ON ATRIB(25).
TASK  ENTER,1,1;
      ACT/1;
```

```
GETR  EVENT,13,1; GET MANPOWER/SE, DETERMINE TASK TIME
      ACT/2,,ATRIB(8).EQ.99,TERM; NO MANPOWER
      ACT/3,,ATRIB(25).NE.2.0,RPR;NOT R2, DONT NEED DELAY OR PART
      ACT/4,,ATRIB(9).EQ.1.0,RPR; R2 ON LATER SHIFT, DONT NEED DELAY
      ACT/5,ATRIB(5);LOGISTICS DELAY, DONT CHANGE ACT#
PART  EVENT,19,2; DEMAND FOR PART
      ACT/20,.5,,SHOP; TRANSPORT LRU TO SHOP
      ACT/6,,ATRIB(10).EQ.0,NOSP; NO SPARE
      ACT/7,,,RPR;
RPR   GOON,1;
      ACT/8,ATRIB(26);
FREE  EVENT,14,1; RELEASE PEOPLE, ADD MANHOURS
      ACT/9,,ATRIB(8).EQ.99.,TERM; IF DUMMY OR T38
      ACT/99,,ATRIB(1).GT.380.,TERM; IF T38
      ACT/10,.02,ATRIB(26).NE.0,TASK; FINISH TASK ON NEXT SHIFT
      ACT/11,,,NEXT; TASK COMPLETE, FIND NEXT TASK
NEXT  EVENT,20,1; DETERMINE NEXT TASK
      ACT/12,,ATRIB(8).EQ.99,TERM; REPAIR COMPLETED
      ACT/13,,,TASK; NEXT TASK
;END OF TASK NETWORK
;FUNCTIONAL CHECK FLIGHT
FCF   ENTER,4,1; FCF CHECKED IN EVENT 20, IF SUCCESSFUL, SEND TO HOLD
      ACT/19,1.3,,TASK; IF FCF FAILED, REPEAT TASK
;NO SPARE NETWORK, EITHER CANNIBALIZE OR ORDER PART AND WAIT
NOSP  ASSIGN,XX(4)=ATRIB(7),2; LOOKING FOR PART XX(10)
      ACT/14,,,EV9;USE DUMMY TO CHECK, SINCE ATRIBS CHANGE
      ACT/15,.001;
      GOON,1;
      ACT/16,,XX(5).EQ.0,ORDR; MUST ORDER AND WAIT
      ACT;
      ASSIGN,ATRIB(27)=2*ATRIB(27)+ATRIB(26),1;DOUBLE R2 TIME
      ACT/17,ATRIB(26),XX(5).GT.0,FREE;REPAIR TIME
ORDR  EVENT,10,1;
      ACT/18,,,FREE; DUMMY RELEASES PEOPLE
;SHOP
SHOP  ASSIGN,ATRIB(1)=200.0,ATRIB(25)=6,ATRIB(26)=0,1; SET CODES
      ACT,,,TASK; SEND TO TASK NETWORK FOR SHOP REPAIR
;CHECK CANNIBALIZATION
EV9   EVENT,9,1;
      ACT,,,TERM;
TERM  TERM; ALL DUMMY ENTITIES TERMINATED HERE
;****************************************************************
      ENDNETWORK;
;RUN LENGTH 32 WEEKS WITH MTBM=2.9, (IF RUNNING T37, MTBM=2.58).
INIT,0,3000.0;
INTLC,XX(1)=3.2;
MONTR,CLEAR,336.0;
SIMULATE;
FIN;
```

```
      BLOCK DATA T46AX
      COMMON/T46AD1/JDATA(4)
      COMMON/SCENE/PROGRM(4),MXINFO(5,13),SCINFO(5,12),SFTLEN(4)
      COMMON/SYSTM/ACINFO(3),TASKIN(12,5)
      COMMON/WUC/CODES(65,9),JRSCTN(65,12),TIMES(65,12),SHOP(65,5)
*
*****       CODES FOR READING DATA FILE   *****
*
*** JDATA1=# OF RES. TYPES,JDATA2=RES# OF DUMMY,JDATA3=# OF WUC'S,
*** JDATA4=WUC OF ENGINE
      DATA JDATA/13,12,62,15/
*****         SCENARIO LEVEL INPUTS       *****
*
*** PROGRM1=#OF A/C, 2=SCHDL TO FLY, 3=SPARES, 4= UTE RATE
      DATA PROGRM/108,53,9,60/
*** SFTLEN = LENGTH OF EACH SHIFT - DAY, EVEN, NIGHT, WEEKEND
      DATA SFTLEN/9,8,7,48/
*** MXINFO(ISHIFT,JRSC) IS # OF JTH RESOURCE ON ISHIFT
*** MXINFO(4,JRSC) IS THE PRODUCTIVITY FACTOR FOR THAT RESOURCE
*** MXINFO(5,JRSC) IS THE # OF ANNUAL MANHOURS SPENT ON T38 BY RSC
*** RSCS PER 24FEB87 MSG=25,24,27,12,16,12,25,68,3,45,164,199,1/
*** (HAVE TO ADD EGRESS WHEN SHARING)
*** INST,COMNAV,ELEC,ECS,EGRS,FUEL,HYD,ENG,DOCK,SHT/MCH,APG,DUM,AUTOECS
*** NOTE: ADD 5.5 PEOPLE/PHASE DOCK/SHIFT; ADD 1.5 PEOPLE/AUTOECS/SHIFT
      DATA (MXINFO(1,J),J=1,13)/12,13,15,6,12,6,13,35,3,24,80,199,1/
      DATA (MXINFO(2,J),J=1,13)/13,11,12,6,12,6,12,35,3,24,70,199,1/
      DATA (MXINFO(3,J),J=1,13)/0,0,0,0.0,0,0,0,0,0,15,199,1/
      DATA (MXINFO(4,J),J=1,13)/76,66,72,72,72,65,74,76,100,62
     1,76,100,100/
      DATA (MXINFO(5,J),J=1,13)/19135,15387,20666,9676,24327,4755
     1,18053,0,0,22876,0,0,0/
*** USED FOR DEDICATED RESOURCES   DATA (MXINFO(5,J),J=1,13)/13*0/
*** SCINFO1 % TOTAL ANNUAL FH/MONTH; SCINFO2, FLYING DAYS/MONTH
*** SCINFO3 = % SORTIES CANCELLED DUE TO WEATHER
*** SCINFO4 AND 5 IS SUNRISE-30MIN AND SUNSET+30MIN
      DATA (SCINFO(1,J),J=1,12)/7.4,6.6,8.6,8.6,8.4,9.1,9.5
     1,10.6,8.5,8.5,8.2,5.7/
      DATA (SCINFO(2,J),J=1,12)/22,19,21,22,23,20,22,22,20,23,20,20/
      DATA (SCINFO(3,J),J=1,12)/28,29,25,22,22,21,14,08,14,18,17,18/
      DATA (SCINFO(4,J),J=1,12)/700,645,630,600,600,600,600,600
     1,610,620,630,645/
      DATA (SCINFO(5,J),J=1,12)/1805,1830,1900,1940,2005,2030
     1,2015,2000,1925,1845,1810,1800/
*
*****        SYSTEM LEVEL INPUTS      *****
*
***ACINFO1=FH BETWEEN AIRFRAME PHASE, ACINFO2=FH BETWEEN ENGINE PHASE
***ACINFO3=TASK TIME FOR PAINT
      DATA ACINFO/600,900,96/
```

```
***TASKIN (1=ENGINE PHASE, 2=AUTOECS, 3=TEST CELL, 4=MAJOR PHASE,
***  5=MINOR PHASE, 6=TOW IN, 7=PREFLIGHT, 8=THRU-FLIGHT SERVICING,
***  9=POSTFLIGHT, 10=MONTHLY SERVICE, 11=200FH SERVICE, 12=TOW OUT)
      DATA (TASKIN(1,J),J=1,5)/8,3,0,0,53/
      DATA (TASKIN(2,J),J=1,5)/13,1,0,0,14.5/
      DATA (TASKIN(3,J),J=1,5)/8,3,0,0,2.33/
      DATA (TASKIN(4,J),J=1,5)/9,1,0,0,80.0/
      DATA (TASKIN(5,J),J=1,5)/9,1,0,0,48.0/
      DATA (TASKIN(6,J),J=1,5)/11,4,0,0,1.0/
      DATA (TASKIN(7,J),J=1,5)/11,1,0,0,.42/
      DATA (TASKIN(8,J),J=1,5)/11,1,0,0,.20/
      DATA (TASKIN(9,J),J=1,5)/11,1,0,0,.75/
      DATA (TASKIN(10,J),J=1,5)/11,1,0,0,4.0/
      DATA (TASKIN(11,J),J=1,5)/11,1,0,0,0.65/
      DATA (TASKIN(12,J),J=1,5)/11,4,0,0,1.0/
*
*****       WUC LEVEL INPUTS     ******
*
*** CODES
*    1 - MTBM                       5 - % FUNCTIONAL CHECK
*    2 - CANNIBALIZATION CODE       6 - % FCF NEEDED
*    3 - % NMCS                     7 - % CND
*    4 - % ON EQUIPMENT REPAIR      8 - % REJECTS
*                                   9 - % TOWED OR JACKED
*** JRSCTN - RESOURCE TYPE AND NUMBER
*     1,3 RESR CODES;     2,4 # TO TROUBLESHOOT
*     5,7 RESR CODES;     6,8 # TO REPAIR
*    9,11 RESR CODES;  10,12 # TO FUNCTIONAL (OPS) CHECK
*** AFSC  CODE                 AFSC  CODE
*   325X1   1   INSTRUMENTS    423X4   7    HYDRAULICS
*   328X0   2   COMNAV         426X2   8    ENGINES
*   423X0   3   ELECTRICS      427X1   9    CORR
*   423X1   4   ECS            427X5  10    MACH,WELD,SHTMTL
*   423X2   5   EGRESS         APG    11    CREW CHIEF
*   423X3   6   FUEL           DUMMY  12    DUMMY
*                              AUTOECS13    AUTO ECS
*** TIMES - MIN, MOST LIKELY, AND MAX TIMES (IN MINUTES)
*      1,2,3  TROUBLESHOOT
*      4,5,6  REPAIR ON AIRCRAFT
*      7,8,9  REMOVE AND REPLACE
*   10,11,12  FUNCTIONAL CHECK
*** SHOP
*     1 - RESR TYPE   2 - # OF RESOURCES
*    3,4,5 - LOW, MOST LIKELY, HIGH TIME IN MINUTES FOR REPAIR
*** WUC 11000, EXCEPT 11ACA, MAINTENANCE BY APG(50%)
      DATA(CODES(1,J),J=1,9)/35,1,.05,.9,0,0,0,0,.5/
      DATA(JRSCTN(1,J),J=1,12)/10,1,11,1,11,1,0,0,0,0,0,0/
      DATA(TIMES(1,J),J=1,12)/5,10,20,10,15,30,10,15,30,3*0/
      DATA(SHOP(1,J),J=1,5)/10,1,113,134,197/
*** WUC 11000, EXCEPT 11ACA, MAINTENANCE BY SHEETMETAL(50%)
      DATA(CODES(2,J),J=1,9)/35,1,.05,.7,0,0,0,0,.5/
      DATA(JRSCTN(2,J),J=1,12)/10,1,11,1,10,2,0,0,0,0,0,0/
      DATA(TIMES(2,J),J=1,12)/5,10,20,15,25,150,13,50,190,3*0/
      DATA(SHOP(2,J),J=1,5)/10,1,113,134,197/
```

```
*** WUC 11ACA  WINDSHIELD TRANSPARENCY (MAX=1.5*LSA TIME)
      DATA(CODES(3,J),J=1,9)/152,1,.05,1,1,0,0,0,.5/
      DATA(JRSCTN(3,J),J=1,12)/11,1,10,1,11,1,10,1,11,1,10,1/
      DATA(TIMES(3,J),J=1,12)/5,10,15,3*0,280,320,480,20,30,45/
      DATA(SHOP(3,J),J=1,5)/10,1,113,134,197/
*** WUC 12000, EXCEPT 12BDB,12DBA,12BF0,12C00
      DATA(CODES(4,J),J=1,9)/113,1,.05,0,0,0,.01,.1,0/
      DATA(JRSCTN(4,J),J=1,12)/11,2,0,0,11,2,0,0,11,1,0,0/
      DATA(TIMES(4,J),J=1,12)/0,10,45,3*0,15,45,60,3*0/
      DATA(SHOP(4,J),J=1,5)/3,1,34,40,59/
*** WUC 12BDB CANNOPY ACTUATOR
      DATA(CODES(5,J),J=1,9)/153,1,.05,0,0,0,.01,.1,0/
      DATA(JRSCTN(5,J),J=1,12)/11,1,0,0,11,1,0,0,11,1,0,0/
      DATA(TIMES(5,J),J=1,12)/10,30,60,3*0,45,60,180,3*0/
      DATA(SHOP(5,J),J=1,5)/3,1,34,40,59/
*** WUC 12000, CIRCUITS, RELAYS AND SWITCHES
      DATA(CODES(6,J),J=1,9)/952,1,.05,1,0,0,.01,.1,0/
      DATA(JRSCTN(6,J),J=1,12)/3,1,0,0,3,1,0,0,4*0/
      DATA(TIMES(6,J),J=1,12)/20,45,60,5,20,45,6*0/
      DATA(SHOP(6,J),J=1,5)/3,1,34,40,59/
*** WUC 12BDA, CANOPY ACTUATOR REMOVER ASSY
      DATA(CODES(7,J),J=1,9)/233,1,.05,0,1,0,.01,1,0/
      DATA(JRSCTN(7,J),J=1,12)/11,2,3,1,5,2,11,1,11,2,3,1/
      DATA(TIMES(7,J),J=1,12)/30,60,90,3*0,120,180,210,30,60,90/
      DATA(SHOP(7,J),J=1,5)/3,1,34,40,59/
*** WUC 12BF0, CANOPY EMERGENCY RELEASE SYS
      DATA(CODES(8,J),J=1,9)/9259,1,.05,0,0,0,.01,.1,0/
      DATA(JRSCTN(8,J),J=1,12)/4*0,5,2,0,0,4*0/
      DATA(TIMES(8,J),J=1,12)/6*0,30,60,150,3*0/
      DATA(SHOP(8,J),J=1,5)/3,1,34,40,59/
*** WUC 12C00; ACESII SEAT (MAX=9 HOURS)
      DATA(CODES(9,J),J=1,9)/112,0,.05,.5,1,0,.01,.1,0/
      DATA(JRSCTN(9,J),J=1,12)/5,2,3,1,5,2,11,1,5,2,0,0/
      DATA(TIMES(9,J),J=1,12)/30,40,50,30,45,90,60,85,120,5,6,10/
      DATA(SHOP(9,J),J=1,5)/5,2,30,50,85/
*** WUC 13AL0/BU0, TIRES
      DATA(CODES(10,J),J=1,9)/59,0,.05,.95,0,0,.15,0,0/
      DATA(JRSCTN(10,J),J=1,12)/4*0,11,1,0,0,4*0/
      DATA(TIMES(10,J),J=1,12)/3*0,10,15,30,10,15,30,3*0/
      DATA(SHOP(10,J),J=1,5)/10,1,97,115,170/
*** WUC 13000, LANDING GEAR EXCEPT 13BP0 AND TIRES
      DATA(CODES(11,J),J=1,9)/94,1,.05,.8,1,0,.15,.05,.55/
      DATA(JRSCTN(11,J),J=1,12)/7,1,11,1,7,1,11,1,7,1,11,1/
      DATA(TIMES(11,J),J=1,12)/5,7,10,45,55,70,15,25,40,10,15,20/
      DATA(SHOP(11,J),J=1,5)/10,1,120,155,180/
*** WUC 14000, FLIGHT CONTROLS EXCEPT 14DCA,ECB,FCB,FCD,14BC0,CC0
      DATA(CODES(12,J),J=1,9)/96,1,.05,.75,1,0.25,.08,0/
      DATA(JRSCTN(12,J),J=1,12)/11,2,3,1,11,2,0,0,11,2,3,1/
      DATA(TIMES(12,J),J=1,12)/45,60,120,45,90,360,45,90,360,60,90,120/
      DATA(SHOP(12,J),J=1,5)/3,1,106,125,185/
*** WUC 14DCA,ECB,FCB,FCD; ACTUATORS
      DATA(CODES(13,J),J=1,9)/598,1,.05,0,1,0,.25,.05,0/
      DATA(JRSCTN(13,J),J=1,12)/7,1,11,1,7,1,0,0,7,1,11,1/
      DATA(TIMES(13,J),J=1,12)/45,70,90,3*0,45,70,90,30,45,60/
      DATA(SHOP(13,J),J=1,5)/7,1,106,125,185/
```

```
*** WUC 14BC0,14CC0, ROLL PITCH ELEC COMP
      DATA(CODES(14,J),J=1,9)/839,1,.05,0,1,0,.25,.08,0/
      DATA(JRSCTN(14,J),J=1,12)/3,2,11,1,3,1,0,0,11,2,0,0/
      DATA(TIMES(14,J),J=1,12)/25,30,35,3*0,40,50,60,10,15,20/
      DATA(SHOP(14,J),J=1,5)/3,1,106,125,185/
*** WUC 23000, ENGINE CORE
*** DOES NOT INCLUDE 23TA0,23TB0,23TE0,23L00,23SA0,23TAA,AB,BA,BB
      DATA(CODES(15,J),J=1,9)/100,1,.05,0,1,0,0,0,.5/
      DATA(JRSCTN(15,J),J=1,12)/8,2,0,0,8,3,0,0,8,1,11,1/
      DATA(TIMES(15,J),J=1,12)/15,30,60,3*0,60,90,200,15,30,60/
      DATA(SHOP(15,J),J=1,5)/8,4,106,125,185/
*** WUC 23TA0, MASTER QUANDRANT ASSY
      DATA(CODES(16,J),J=1,9)/427,0,.05,.3,1,0,0,.01,0/
      DATA(JRSCTN(16,J),J=1,12)/3,1,11,1,11,1,3,2,11,2,0,0/
      DATA(TIMES(16,J),J=1,12)/5,30,60,180,240,300,180,240,300
     1,10,30,60/
      DATA(SHOP(16,J),J=1,5)/8,2,106,125,185/
*** WUC 23TB0, SLAVE QUANDRANT ASSY
      DATA(CODES(17,J),J=1,9)/743,0,.05,0,1,0,0,.01,0/
      DATA(JRSCTN(17,J),J=1,12)/11,1,3,1,11,1,3,2,11,2,0,0/
      DATA(TIMES(17,J),J=1,12)/15,30,60,3*0,180,240,300,15,30,60/
      DATA(SHOP(17,J),J=1,5)/8,2,106,125,185/
*** WUC 23L00,23SA0,23TAA,AB,BA,BB
      DATA(CODES(18,J),J=1,9)/491,1,.05,0,1,0,0,.01,0/
      DATA(JRSCTN(18,J),J=1,12)/3,2,0,0,3,2,0,0,3,2,11,1/
      DATA(TIMES(18,J),J=1,12)/45,60,90,3*0,15,20,30,15,20,30/
      DATA(SHOP(18,J),J=1,5)/3,1,40,45,55/
*** WUC 23TE0, ECU
      DATA(CODES(19,J),J=1,9)/167,0,.05,.5,1,1,0,0,0/
      DATA(JRSCTN(19,J),J=1,12)/1,1,0,0,1,1,8,1,8,2,11,1/
      DATA(TIMES(19,J),J=1,12)/10,30,60,50,65,90,50,65,80,10,15,20/
      DATA(SHOP(19,J),J=1,5)/3,1,25,30,40/
*** WUC 23000, ENGINE
      DATA(CODES(20,J),J=1,9)/398,1,.05,1,1,0,0,.01,0/
      DATA(JRSCTN(20,J),J=1,12)/8,2,0,0,8,1,0,0,8,2,11,1/
      DATA(TIMES(20,J),J=1,12)/15,30,45,30,60,90,15,30,45,10,15,20/
      DATA(SHOP(20,J),J=1,5)/3,1,162,192,283/
*** WUC 41EBA, ICE DETECTOR
      DATA(CODES(21,J),J=1,9)/595,1,.05,0,1,0,0,0,0/
      DATA(JRSCTN(21,J),J=1,12)/4,2,0,0,4,2,0,0,4,1,11,2/
      DATA(TIMES(21,J),J=1,12)/15,20,30,3*0,20,30,45,15,30,45/
      DATA(SHOP(21,J),J=1,5)/4,1,157,186,275/
*** WUC 41DC0,41EB0,41F00
      DATA(CODES(22,J),J=1,9)/269,1,.05,0,0,0,0,0,0/
      DATA(JRSCTN(22,J),J=1,12)/3,1,0,0,3,1,6*0/
      DATA(TIMES(22,J),J=1,12)/15,25,45,3*0,20,45,105,3*0/
      DATA(SHOP(22,J),J=1,5)/4,1,157,186,275/
*** WUC 41000, PRESS.& AIR COND, EXCEPT 41DC0,41EB0,41F00,41EBA
** AND AIR CONDITIONING PACKAGE(41BA0)
      DATA(CODES(23,J),J=1,9)/86,1,.05,0,1,0,0,.02,0/
      DATA(JRSCTN(23,J),J=1,12)/4,1,11,1,4,1,0,0,4,1,11,1/
      DATA(TIMES(23,J),J=1,12)/5,10,15,10,15,20,20,30,45,10,15,25/
      DATA(SHOP(23,J),J=1,5)/4,1,157,186,275/
```

```
*** WUC 42AA0, AND 42ED0, STARTER GENERATOR AND ENG RELAY BOX
     DATA(CODES(24,J),J=1,9)/589,1,.05,0,1,0,.05,0,0/
     DATA(JRSCTN(24,J),J=1,12)/3,2,11,1,11,2,0,0,11,2,0,0/
     DATA(TIMES(24,J),J=1,12)/10,15,25,0,0,0,150,165,200,10,15,18/
     DATA(SHOP(24,J),J=1,5)/3,1,110,130,197/
*** WUC 42CA0, BATTERY ASSY
     DATA(CODES(25,J),J=1,9)/376,1,.05,0,0,0,.05,.05,0/
     DATA(JRSCTN(25,J),J=1,12)/3,1,11,1,11,1,6*0/
     DATA(TIMES(25,J),J=1,12)/10,15,20,3*0,15,20,30,3*0/
     DATA(SHOP(25,J),J=1,5)/3,1,110,130,197/
*** WUC 42000, ELECTRICAL POWER SUPPLY EXCEPT 42AA0, 42CA0, 42ED0
     DATA(CODES(26,J),J=1,9)/253,1,.05,0,1,0,.05,.1,0/
     DATA(JRSCTN(26,J),J=1,12)/3,1,11,1,3,1,0,0,3,1,11,1/
     DATA(TIMES(26,J),J=1,12)/3,6,9,0,0,0,20,40,60,5,10,20/
     DATA(SHOP(26,J),J=1,5)/3,1,110,130,197/
*** WUC 44000, LIGHTING
     DATA(CODES(27,J),J=1,9)/131,1,.05,0,1,0,.04,.04,0/
     DATA(JRSCTN(27,J),J=1,12)/3,1,11,1,3,1,0,0,3,1,11,1/
     DATA(TIMES(27,J),J=1,12)/5,10,20,3*0,10,18,25,5,10,15/
     DATA(SHOP(27,J),J=1,5)/3,1,55,65,97/
*** WUC 45000, HYD PNEUMATIC POWER SUPPLY
     DATA(CODES(28,J),J=1,9)/245,1,.05,0,1,0,.3,.01,.1/
     DATA(JRSCTN(28,J),J=1,12)/7,2,0,0,7,2,0,0,7,2,8,1/
     DATA(TIMES(28,J),J=1,12)/10,30,60,3*0,15,50,120,15,30,60/
     DATA(SHOP(28,J),J=1,5)/7,1,162,192,283/
*** WUC 45CA0, HYD PRESSURE INDICATING SYS
     DATA(CODES(29,J),J=1,9)/1504,0,.05,1,0,0,.3,.01,.01/
     DATA(JRSCTN(29,J),J=1,12)/3*0,0,1,1,6*0/
     DATA(TIMES(29,J),J=1,12)/3*0,30,60,120,3*0,3*0/
     DATA(SHOP(29,J),J=1,5)/7,1,162,192,283/
*** WUC 46000, FUEL SYS EXCEPT 46CA0, 46ECB,46DAB
     DATA(CODES(30,J),J=1,9)/373,0,.05,1,0,0,.02,.01,.1/
     DATA(JRSCTN(30,J),J=1,12)/4*0,6,2,0,0,4*0/
     DATA(TIMES(30,J),J=1,12)/3*0,40,50,65,3*0,3*0/
     DATA(SHOP(30,J),J=1,5)/6,2,160,190,280/
*** WUC 46CA0, FUEL QUANTITY INDICATING SYS
     DATA(CODES(31,J),J=1,9)/1058,0,.05,0,1,0,.02,.01,.95/
     DATA(JRSCTN(31,J),J=1,12)/1,1,11,1,1,1,6,2,1,1,11,1/
     DATA(TIMES(31,J),J=1,12)/5,10,30,3*0,50,60,90,15,20,35/
     DATA(SHOP(31,J),J=1,5)/6,2,160,190,280/
*** WUC 47000, OXYGEN SYS
     DATA(CODES(32,J),J=1,9)/135,1,.05,.5,1,0,0,.02,0/
     DATA(JRSCTN(32,J),J=1,12)/4,1,0,0,4,1,0,0,4,1,0,0/
     DATA(TIMES(32,J),J=1,12)/15,30,45,20,30,45,20,30,45,15,30,60/
     DATA(SHOP(32,J),J=1,5)/4,1,35,42,62/
*** WUC 47AC0, LOX QUANTITY INDICATING SYS
     DATA(CODES(33,J),J=1,9)/1587,0,.05,1,0,0,0,0,0/
     DATA(JRSCTN(33,J),J=1,12)/3*0,0,1,1,6*0/
     DATA(TIMES(33,J),J=1,12)/3*0,30,60,120,6*0/
     DATA(SHOP(33,J),J=1,5)/4,1,35,42,62/
*** WUC 49000, MISC UTILITIES (FIRE DETECTION)
     DATA(CODES(34,J),J=1,9)/409,1,.05,.5,0,0,0,0,0/
     DATA(JRSCTN(34,J),J=1,12)/4*0,3,1,0,0,3,1,0,0/
     DATA(TIMES(34,J),J=1,12)/3*0,30,40,55,10,45,90,5,6,10/
     DATA(SHOP(34,J),J=1,5)/3,2,120,150,200/
```

```
*** WUC 51000, INSTRUMENTS EXCEPT D00,AA0,BA0,CA0
      DATA(CODES(35,J),J=1,9)/72,1,.05,.8,1,0,.02,.01,0/
      DATA(JRSCTN(35,J),J=1,12)/1,2,0,0,1,1,0,0,1,2,0,0/
      DATA(TIMES(35,J),J=1,12)/15,30,60,20,30,45,10,20,30,15,30,60/
      DATA(SHOP(35,J),J=1,5)/1,1,67,80,118/
*** WUC 51D00, PITOT STATIC SYS
      DATA(CODES(36,J),J=1,9)/553,0,.05,.9,0,.02,.01,0/
      DATA(JRSCTN(36,J),J=1,12)/4*0,1,2,0,0,4*0/
      DATA(TIMES(36,J),J=1,12)/3*0,45,60,240,45,60,80,3*0/
      DATA(SHOP(36,J),J=1,5)/1,2,100,120,150/
*** WUC 51AA0, ADI ARU-39/A
      DATA(CODES(37,J),J=1,9)/66,1,.05,.9,0,0,.02,.01,0/
      DATA(JRSCTN(37,J),J=1,12)/4*0,1,1,0,0,4*0/
      DATA(TIMES(37,J),J=1,12)/3*0,25,35,50,15,30,90,3*0/
      DATA(SHOP(37,J),J=1,5)/1,2,100,120,150/
*** WUC 51BA0, HSI AQU-6/A
      DATA(CODES(38,J),J=1,9)/118,1,.05,0,1,0,.02,.01,0/
      DATA(JRSCTN(38,J),J=1,12)/1,1,2,1,1,1,0,0,1,1,2,1/
      DATA(TIMES(38,J),J=1,12)/15,30,45,3*0,5,10,20,10,15,30/
      DATA(SHOP(38,J),J=1,5)/1,1,67,80,100/
*** WUC 51CA0, DISPLACEMENT GYRO AN/ASN-129A
      DATA(CODES(39,J),J=1,9)/323,1,.05,.8,1,0,.02,.01,0/
      DATA(JRSCTN(39,J),J=1,12)/4*0,1,1,0,0,1,1,0,0/
      DATA(TIMES(39,J),J=1,12)/3*0,20,30,60,20,30,60,10,15,16/
      DATA(SHOP(39,J),J=1,5)/1,2,180,200,240/
*** WUC 52000, AUTO PILOT SYS
      DATA(CODES(40,J),J=1,9)/421,1,.05,.5,1,0,0,0,0/
      DATA(JRSCTN(40,J),J=1,12)/1,2,0,0,1,1,0,0,1,2,0,0/
      DATA(TIMES(40,J),J=1,12)/15,30,60,10,20,35,5,10,20,15,30,45/
      DATA(SHOP(40,J),J=1,5)/1,1,150,180,210/
*** WUC 55000, ADR EXCEPT 55AA0
      DATA(CODES(41,J),J=1,9)/10870,0,.05,1,0,0,.01,.1,0/
      DATA(JRSCTN(41,J),J=1,12)/3*0,0,1,1,0,0,4*0/
      DATA(TIMES(41,J),J=1,12)/3*0,30,60,150,3*0,3*0/
      DATA(SHOP(41,J),J=1,5)/3,1,240,320,480/
*** WUC 55AA0, ADR RECORDER
      DATA(CODES(42,J),J=1,9)/569,1,.05,.1,0,0,.01,.1,0/
      DATA(JRSCTN(42,J),J=1,12)/4*0,1,1,0,0,4*0/
      DATA(TIMES(42,J),J=1,12)/3*0,20,30,45,15,30,60,3*0/
      DATA(SHOP(42,J),J=1,5)/3,1,240,320,480/
*** WUC 62000, VHF COMM EXCEPT 62AA0,62AC0
      DATA(CODES(43,J),J=1,9)/11904,0,.05,.9,0,0,.01,0,0/
      DATA(JRSCTN(43,J),J=1,12)/4*0,1,1,0,0,4*0/
      DATA(TIMES(43,J),J=1,12)/3*0,5,10,60,25,35,50,3*0/
      DATA(SHOP(43,J),J=1,5)/2,1,120,180,300/
*** WUC 62AA0, RADIO RECEIVER TRANSMITTER
      DATA(CODES(44,J),J=1,9)/220,1,.05,.2,0,.01,0,0/
      DATA(JRSCTN(44,J),J=1,12)/2,1,0,0,2,1,6*0/
      DATA(TIMES(44,J),J=1,12)/1,2,5,5,10,15,20,30,45,3*0/
      DATA(SHOP(44,J),J=1,5)/2,2,190,230,340/
*** WUC 62AC0, VHF/AM COMM VOR/ILS DUAL CNTRL
      DATA(CODES(45,J),J=1,9)/425,1,.05,.1,1,0,.01,0,0/
      DATA(JRSCTN(45,J),J=1,12)/2,1,0,0,2,1,6*0/
      DATA(TIMES(45,J),J=1,12)/1,2,5,5,10,15,35,45,60,3*0/
      DATA(SHOP(45,J),J=1,5)/2,1,190,230,280/
```

```
*** WUC 63000, UHF COMM EXCEPT 63AA0
      DATA(CODES(46,J),J=1,9)/1585,1,.05,.5,1,0,.01,0,0/
      DATA(JRSCTN(46,J),J=1,12)/2,1,0,0,2,1,0,0,2,1,0,0/
      DATA(TIMES(46,J),J=1,12)/10,15,45,5,10,25,5,7,20,5,10,15/
      DATA(SHOP(46,J),J=1,5)/2,1,190,230,340/
*** WUC 63AA0, UHF/AM RADIO RT-1l6AB/ARC-164
      DATA(CODES(47,J),J=1,9)/181,1,.05,.3,1,0,.1,.01,0/
      DATA(JRSCTN(47,J),J=1,12)/2,1,0,0,2,1,0,0,2,1,0,0/
      DATA(TIMES(47,J),J=1,12)/5,10,60,30,40,50,5,10,15,2,5,10/
      DATA(SHOP(47,J),J=1,5)/2,1,190,230,280/
*** WUC 64000, INTERPHONE
      DATA(CODES(48,J),J=1,9)/220,1,.05,.5,1,0,.05,.01,0/
      DATA(JRSCTN(48,J),J=1,12)/2,2,0,0,2,1,0,0,2,2,0,0/
      DATA(TIMES(48,J),J=1,12)/10,30,60,5,30,60,5,10,30,10,15,20/
      DATA(SHOP(48,J),J=1,5)/2,1,100,120,180/
*** WUC 65000, IFF EXCEPT  65AA0
      DATA(CODES(49,J),J=1,9)/9690,1,.05,0,1,0,.01,.01,.10/
      DATA(JRSCTN(49,J),J=1,12)/2,2,0,0,2,2,0,0,2,2,1,1/
      DATA(TIMES(49,J),J=1,12)/5,10,30,3*0,8,15,30,3,5,8/
      DATA(SHOP(49,J),J=1,5)/2,1,205,243,358/
*** WUC 65AA0, RT-1426 TRANSPONDER SET
      DATA(CODES(50,J),J=1,9)/252,1,.05,.1,1,0,.01,.01,0/
      DATA(JRSCTN(50,J),J=1,12)/2,2,0,0,2,2,6*0/
      DATA(TIMES(50,J),J=1,12)/1,2,5,5,10,15,20,30,45,3*0/
      DATA(SHOP(50,J),J=1,5)/2,1,60,160,240/
*** WUC 71A00, VOR/ILS/MB SYS AN/ARN-127
      DATA(CODES(51,J),J=1,9)/210,1,.05,.3,1,0,.01,.05,0/
      DATA(JRSCTN(51,J),J=1,12)/2,2,0,0,2,1,0,0,2,2,0,0/
      DATA(TIMES(51,J),J=1,12)/5,10,15,15,25,40,30,45,60,5,10,15/
      DATA(SHOP(51,J),J=1,5)/2,1,200,220,250/
*** WUC 71BA0, RT-1159/A RECEIVER/TRANSMITTER
      DATA(CODES(52,J),J=1,9)/539,1,.05,.4,0,0,.01,.05,0/
      DATA(JRSCTN(52,J),J=1,12)/4*0,2,2,0,0,4*0/
      DATA(TIMES(52,J),J=1,12)/3*0,25,35,60,30,45,60,3*0/
      DATA(SHOP(52,J),J=1,5)/2,1,150,170,200/
*** WUC 71BD0, TACAN CONTROL PANEL
      DATA(CODES(53,J),J=1,9)/614,1,.05,.1,1,0,.01,.05,0/
      DATA(JRSCTN(53,J),J=1,12)/2,2,0,0,2,1,0,0,2,2,0,0/
      DATA(TIMES(53,J),J=1,12)/5,15,30,10,15,30,5,10,15,5,10,15/
      DATA(SHOP(53,J),J=1,5)/2,1,90,100,110/
*** WUC 71BC0, MOUNT NT-4915/A TACAN RT MOUNT
      DATA(CODES(54,J),J=1,9)/637,0,.05,1,0,0,.01,.05,0/
      DATA(JRSCTN(54,J),J=1,12)/3*0,0,2,1,6*0/
      DATA(TIMES(54,J),J=1,12)/3*0,5,25,70,6*0/
      DATA(SHOP(54,J),J=1,5)/2,2,20,30,60/
*** WUC 91000, EMERGENCY EQUIP EXCEPT 91AD0
      DATA(CODES(55,J),J=1,9)/4782,0,.05,0,0,0,.01,.05,0/
      DATA(JRSCTN(55,J),J=1,12)/4*0,2,2,0,0,4*0/
      DATA(TIMES(55,J),J=1,12)/6*0,100,120,150,3*0/
      DATA(SHOP(55,J),J=1,5)/3,1,50,60,75/
*** WUC 91AD0, EMERGENCY OXYGEN ASSY
      DATA(CODES(56,J),J=1,9)/2020,0,.05,1,0,0,.01,.05,0/
      DATA(JRSCTN(56,J),J=1,12)/4*0,11,2,0,0,4*0/
      DATA(TIMES(56,J),J=1,12)/3*0,3*0,100,120,140,3*0/
      DATA(SHOP(56,J),J=1,5)/3,1,0,0,0/
```

```
*** WUC 13BPO, NOSE LANDING GEAR STRUT,(USES TOWING FOR JACKING)
      DATA(CODES(57,J),J=1,9)/3286,1,.05,.3,1,0,.15,.05,1/
      DATA(JRSCTN(57,J),J=1,12)/11,1,7,1,7,2,0,0,11,1,7,1/
      DATA(TIMES(57,J),J=1,12)/5,7,10,45,75,90,80,90,120,10,15,25/
      DATA(SHOP(57,J),J=1,5)/7,1,97,115,170/
*** WUC 14BAO, AILERON
      DATA(CODES(58,J),J=1,9)/1057,1,.05,.75,1,0.25,.08,0/
      DATA(JRSCTN(58,J),J=1,12)/11,1,3,1,11,1,0,0,11,1,3,1/
      DATA(TIMES(58,J),J=1,12)/20,35,50,45,90,360,90,150,300,15,20,30/
      DATA(SHOP(58,J),J=1,5)/3,1,106,125,185/
*** WUC 41BAO, AIR CONDITIONING PACKAGE
      DATA(CODES(59,J),J=1,9)/793,1,.05,0,1,0,0,.02,0/
      DATA(JRSCTN(59,J),J=1,12)/4,1,11,1,4,1,0,0,4,1,11,1/
      DATA(TIMES(59,J),J=1,12)/5,10,15,10,15,20,320,360,400,10,15,25/
      DATA(SHOP(59,J),J=1,5)/4,1,157,186,275/
*** WUC 44BCA, DC POWER CONVERTER
      DATA(CODES(60,J),J=1,9)/9259,1,.05,0,1,0,.04,.04,0/
      DATA(JRSCTN(60,J),J=1,12)/3,2,11,1,3,1,0,0,3,1,11,1/
      DATA(TIMES(60,J),J=1,12)/45,60,90,3*0,100,120,150,5,10,15/
      DATA(SHOP(60,J),J=1,5)/3,1,55,65,97/
*** WUC 46DAB, 46ECB FUEL FEED TUBE, REFUEL SHUTOFF VALVE
      DATA(CODES(61,J),J=1,9)/32680,0,.05,0,1,0,.02,.01,.95/
      DATA(JRSCTN(61,J),J=1,12)/1,1,11,1,1,1,6,2,1,1,11,1/
      DATA(TIMES(61,J),J=1,12)/5,10,20,3*0,220,240,300,15,20,35/
      DATA(SHOP(61,J),J=1,5)/6,2,160,190,280/
*** WUC 46AAA, BA,BB,DCA,EAG; TANKS AND CELLS
      DATA(CODES(62,J),J=1,9)/5765,0,.05,.8,1,0,.02,.01,1/
      DATA(JRSCTN(62,J),J=1,12)/6,1,11,1,6,2,0,0,6,1,11,1/
      DATA(TIMES(62,J),J=1,12)/5,10,20,300,350,420,400,460,550
     1,15,20,35/
      DATA(SHOP(62,J),J=1,5)/6,2,150,500,1000/
      END
```

T46A EXEC

```
&CONTROL OFF
&TYPE RUN USER CODE AND SLAM
FORTVS T46A
FORTVS T46AD
FILEDEF FT20F001 TERM
FILEDEF FT05F001 DISK T46A SLAM A
FILEDEF FT06F001 DISK TSLAM LISTING A
FILEDEF FT07F001 DISK OUT FILE7 (RECFM FB LRECL 132 BLOCK 1320)
FILEDEF FT09F001 DISK T46OUT LISTING A
&TYPE LOADING USER CODE AND SLAM
LOAD T46A T46AD SLAM (CLEAR
START *
ERASE T46A LISTING A
ERASE T46A TEXT A
ERASE T46AD LISTING A
ERASE T46AD TEXT A
```

APPENLIX D. LIST OF VARIABLES

I. FORTRAN Variables

Guidelines Used for Naming Variables:

First Letter(s):	Used For:
AVG or AV	averages, over the length of the simulation
I	indices, either for do loops or array locations
J	integer variable names
K	codes (yes,no; days of the week; etc)
N	numbers or counters (for integers)
TOT	total or cumulative values (for real numbers)
X	real variable names
Z	temporary calculations
I-N	integers (uses FORTRAN default variable typing)
A-H,O-Z	real numbers

Variable Name	Description	Used in Subroutines
ACINFO	array containing system level inputs	CHECK2,INTLC
AVGFMC	average FMC rate	STATS
AVGMDT	mean downtime per sortie	CHECK2,STATS
AVGMIS	% of sorties missed for maintenance	STATS
AVGSPR	daily average # of sorties using spare aircraft	STATS
AVGTRN	average number of ac being turned	STATS,TURN
AVMMH	average maintenance manhours per FH	MXFREE,STATS
AVNMCS	average % of aircraft which are NMCS	SPPLY1,STATS
AVWAIT	average # of planes waiting to be preflighted at sunrise	STATS1
ACINFO	array containing system level inputs	CHECK1,FLYING
CKFMC	time to check FMC rate again	DEBUG
CKTIME	checking for failure at this time	CHECK3
CODES	array containing probabilities and other information on LRUs and WUCs	ASSGNO,CHECK3,INTLC, STATS
DELTA	factor used to adjust individual MTBM's to achieve overall MTBM	INTLC,STATS
DWNTM	downtime for current repair	CHECK2
ENDTSK	completion time for the current task	MXSEIZ
ENGPH	flying hours until phase inspection	CHECK1
FHFAIL	flying hours until next failure	CHECK3,INTLC
ILATER	index for scheduling future groups of sorties	FLYING
IPLANE	index used when searching for a specific aircraft #	SPPLY1
IQ	index used with queues	FLYING,FLYEND,MXQCK

Variable Name	Description	Used in Subroutines
IQNEW	index used when changing queues	SHIFTS
IR1,IR2	indices used for resource types	MXSEIZ,MXFREE
IRSCTN	array, how to read JRSCTN array	MXSEIZ
ISHIFT	shift #, used as index	MXSEIZ,MXFREE,MXQCK SHIFTS
ISTART	queue #, used to start search	MXQCK,SHIFTS
ISTOP	queue #, used to stop search	MXQCK,SHIFTS
ITAIL	index for aircraft #	CHECK2,SPPLY1,SPPLY2
ITEM	index, LRU or WUC being repaired	ASSGNO,CHECK2,CHECK3
ITIMES	array, how to read TIMES array	MXSEIZ
JCOUNT	counter	DEBUG
JDATA	array, info on resources and WUC's	INTLC
JLONG	resource type causing longest delay	STATS
JPARTS	array of parts missing, by a/c #	CHECK2,SPPLY1
JRSC	array of # of resources available by shift and by resource type	MXSEIZ,MXFREE,MXQCK
JRSCTN	array, by WUC, containing resource type and number inputs	INTLC,MXSEI
JSPARE	array, by WUC or LRU, of demands on the supply system	SPAREA,SPAREB
JTAIL	array, of parts missing by aircraft	CHECK2,SPPLY1,SPPLY2
JWAIT	avg # planes, LRUs waiting for people	STATS
KDAY	code for day of the week(1=Monday,etc)	CALNDR,FLYING
KFAIL	code for failure found(0=no,1=yes)	CHECK1,CHECK2,CHECK3
KFLY	code for a/c available(0=no,1=yes)	SORTIE
KFROM	code for why subroutine CHECK3 was called(0=preflight,1=postflight)	CHECK1,CHECK2,CHECK3, SORTIE
KMONTH	code for month (1=Jan, 2=Feb,etc)	CALNDR,FLYING,INTLC
KTASK	code for type of task (renumber and list	MXSEIZ
MXINFO	array, containing resource inputs	SHIFTS
NCALLS	counter for # calls of supply system	SPPLY1
NEND	# of days simulated	STATS
NFLOWN	# of sorties comleted	CHECK2,INTLC,SORTIE, STATS
NFLYDY	# of flying days	CALNDR,FLYING,STATS
NKBALL	counter for # kball actions	SPPLY1
NMISS	cumulative # of missed sorties	CALNDR,SORTIE,STATS
NMISS1	# of missed sorties thru previous day	CALNDR
NOBSV	counter, # observations for FMC calc.	STATS1
NPLANE	# of aircraft in system	INTLC, STATS
NR1,NR2	indices for # of resources	MXSEIZ,MXFREE
NRSC	# of resources	INTLC,MXSEIZ,MXQCK, SHIFTS
NSORTS	array of sorties to be flown (1=local annual, 2=today, 3=now, 4=later today, 5=night, 6=xc/wk)	CALNDR,INTLC,FLYING, FLYDM
NSPR	# of sorties using spare aircraft	CHECK3,INTLC,SORTIE, STATS
NTURN	# of aircraft being turned	DEBUG
NWEEK	# of weeks simulated	CALNDR,INTLC

Variable Name	Description	Used in Subroutines
NWORK	array, # of each resources working current shift	SHIFTS
NWUC	# of WUCs or LRUs modeled	CHECK3,INTLC,STATS
NWUC23	# of WUC for the engine core	CHECK1,INTLC
NXCWK	# of cross country sorties each week	FLYING,INTLC
ORDTIM	delivery time when ordering spare part	SPPLY2
PROGRM	array with scenario inputs	CHECK1,INTLC,STATS, DEBUG
PPAINT	probability of painting during phase	CHECK1
RAND	pseudo-random number	ASSGNO,CHECK1,MXSEIZ SHIFTS
SCINFO	array, containing flying schedule and weather inputs	CALNDR,FLYING
SFTCHG	time of next shift change	SHIFTS
SFTLEN	array, length of each shift	SHIFTS
SHOP	array, by WUC, resources & times to perform shop maintenance	MXSEIZ
SORTA	sortie length	SORTIE
SORTB	sortie completion time	SORTIE
TASKIN	array, inputs for the resources & times for system level tasks	CHEC1,CHECK2,INTLC, FLYEND,FLYING,MXSEIZ
TFMC,TFMCP	used to calculate FMC1 rate	STATSO
TFMC1	FMC rate, calculated at sunrise	STATSO
TIMCLR	warmup period, (=TTCLR)	INTLC
TIMFLT	time of sortie	FLYING,FLYPM
TIMES	array, by WUC, of task times	INTLC,MXSEIZ
TOTDT	total accumulated downtime	CHECK2,INTLC
TOTFH	total accumulated flying hours	CHECK2,CHECK3,SORTIE STATS
TOTTRN	total accumulated time ac being turned	INTLC,TURN
TTWAIT	total # of planes awaiting preflight	STATS1
XLONG	length of the longest queue	STATS
XMDTPS	mean downtime per sortie	STATS
XMMH	maintenance manhours for current task	MXFREE
XMTBM	mean time between maintenance (=xx(1))	STATS
XNMCS	current NMCS percentage	SPPLY2
XNMCSO	input NMCS	INTLC,SPPLY2
XNMCS1	output NMCS	STATS
XNWUC	number of WUC's	INTLC
XWAIT	sum of length of queues	STATS
ZA,ZB,ZC	temporary calculations to make later equation more readable	MXSEIZ
ZCLR	same as above	INTLC
ZHOUR	same as above	FLYING
ZFRATE	same as above	INTLC,STATS
ZFH	same as above	INTLC
ZLONG	same as above	STATS
ZSORTS	same as above	CALNDR
ZWX,ZWX1,ZWX2	same as above	CALNDR

II. SLAM Variables

II.1. Global Variables - The values of global variables apply to the entire system, not to a specific entity. They can be set in either the network or the FORTRAN and apply to a state or condition of the entire simulation.

Variable	Description
XX(1)	MTBM, total corrective
XX(2)	shift number
XX(3)	end of current shift
XX(4)	code for part needed
XX(5)	code for cannibalization (1=yes,0=no)
XX(6)	cumulative maintenance manhours

II.2. Attributes - Each entity has its own attribute array which is attached to the entity as it flows through the network (4:92).

Attribute	Description
1	aircraft #, or code for LRU (=200) or engine (=300)
2	inspection time, in flying hours
3	total flying hours
4	time last flown, or scheduled to fly
5	logistics delay time
6	# of parts missing
7	item currently being repaired
8	aircraft status (2=schdl to fly, 3=spare, 4=needed for sortie, 99=dummy)
9	code for part available for remove and replace
10	spare part available
11	on aircraft repair (1=yes,0=no)
12	functional check (1=yes,0=no)
13	functional check flight (1=yes,0=no)
14	cannot duplicate (1=yes,0=no)
15	reject after functional check (1=yes,0=no)
16	towed for repair (1=yes,0=no)
17	day of last scheduled maintenance
18	FH since last scheduled maintenance
19	not used
20	not used
21	first resource type
22	number of first resource
23	second resource type
24	number of second resource
25	code for type of task
26	task time this shift
27	task time remaining for future shift

APPENDIX E. RUN INSTRUCTIONS

I. FILES

The files can have any name, as long as they are properly identified in the EXEC file. The following are the names used in Appendix E. Computer Code and refered to in the documentation.

Files Used: T46A SLAM - contains network
T46A FORTRAN - contains fortran subroutines
T46AD FORTRAN - contains input data
T46A EXEC - controls input and output files
SLAM TEXT - SLAM language

Output Files: TSLAM LISTING - contains standard SLAM output
T46OUT LISTING - contains output as formatted in the
T46A FORTRAN file

II. INPUT FILE DIMENSIONS

Limits on the number of resources, number of WUC's and number of aircraft are set by the dimensions of certain arrays. The current dimensions, array names, and common block locations for the current dimensions are listed below. To change these limits, change the T46AD FORTRAN file and all occurrences in the T46A FORTRAN file.

	Current	Array Dimensions	Located In
Item	Limit	To Change	Common Block(s)
# aircraft	150	JPARTS	USPR
# resource types	13	MXINFO,JRSC,NWORK	SCENE,USFTS
# WUC's	60	CODES,JRSCTN,TIMES, SHOP,JSPARE	USPR,WUC

If making significant increases to any of these values, especially the number of WUC's, it may be necessary to increase NSET and QSET. NSET and QSET is the amount of file space SLAM sets aside before beginning the run. With the limits listed above, 15000 is sufficient. In order to changed the dimensions of NSET and QSET, there are three statements in PROGRAM MAIN of T46A FORTRAN which need to be changed - DIMENSION NSET(15000), COMMON QSET(15000), and NNSET(15000).

III. INPUTS

The T46AD FORTRAN file contains all of the inputs for the model in a block data format. The inputs can be included in any order; for the description below, the inputs have been divided into three categories - data file inputs, scenario level inputs, system level inputs, and WUC/LRU level inputs. Each of the input arrays are described below, followed by an example input for that array. The sample input does not represent any specific system. Appendix E. Computer Code contains the current inputs for the T-46.

Data File Inputs: Data file inputs relate to how to read the data file. They are input in an array called JDATA.

JDATA(1) = # of resource types
JDATA(2) = # resource type used as dummy
JDATA(3) = # of WUC's
JDATA(4) = # WUC used for engine core

Ex: DATA JDATA/6,5,10,2/

Scenario Inputs: Scenario level inputs relate to the operational environment, rather than the specific system being modeled. There are three arrays which contain scenario level inputs - PROGRM, MXINFO, and SCINFO.

The PROGRM array is a 1x4 array with the following information:

PROGRM(1) = # of aircraft assigned at location
PROGRM(2) = # of aircraft on the daily flying schedule
PROGRM(3) = # of aircraft used as spares, daily
PROGRM(4) = flying hours/month/aircraft (UTE rate)

Ex: DATA PROGRM/50,25,5,60/

The MXINFO array is a 5xN array, where N is the number of different resources used (currently, there are 13 resources). The array contains the following information:

MXINFO(1,JRSC) = # of resources assigned to shift 1, by resource type
MXINFO(2,JRSC) = same as above, for shift 2
MXINFO(3,JRSC) = same as above, for shift 3
MXINFO(4,JRSC) = productivity factor (percentage of time that resource is available for direct labor - 80% input as 80), by resource type
MXINFO(5,JRSC) = the # of annual manhours spent on the T-38

```
Ex:  DATA (MXINFO(1,J), J=1,6)/8,6,7,4,50,11/
     DATA (MXINFO(4,J), J=1,6)/80,70,80,65,100,100/
     DATA (MXINFO(5,J), J=1,6)/0,0,46778,0,9655,0/
```

Note: MXINFO(2,3) is the number of resource 6 assigned to the second shift. The number of that resource actually working on that shift for a particular day is calculated in the model using the productivity factor for that resource (MXINFO(4,3)). MXINFO(5,J) indicates that resources 1,2,4, and 6 are dedicated to this system; 3 and 5 are shared with another system which used 46778 and 9655 annual manhours, respectively.

The SCINFO array is a 5x12 array containing schedule information by month, as follows:

SCINFO(1,MN) = % of annual flying done in each month
SCINFO(2,MN) = # of flying days in each month
SCINFO(3,MN) = % of sorties cancelled due to weather, by month
SCINFO(4,MN) = time of sunrise (-30 min), by month
SCINFO(5,MN) = time of sunset (+30 min), by month

```
Ex:  DATA (SCINFO(1,MN),MN=1,12)/7.4,6.6,8.6,8.6,8.4,9.1,9.5,etc.
     DATA (SCINFO(2,MN),MN=1,12)/22,19,21,22,23,20,22,22,23,20,20/
     DATA (SCINFO(3,MN),MN=1,12)/28,29,25,22,22,21,14,8,14,18,17,18/
     DATA (SCINFO(4,MN),MN=1,12)/700,645,630,600,600,600,600,600,etc.
     DATA (SCINFO(5,MN),MN=1,12)/1805,1830,1900,1940,2005,2030,etc.
```

Note: SCINFO (1,1) is the percentage of annual flying which is done in Jan; SCINFO(1,2), Feb; etc. The percentages for the 12 months should total 100.0.

System Inputs: There are two arrays which contain system level information - ACINFO and TASKIN.

ACINFO is an array with system level information.

ACINFO(1) = FH between airframe phase inspections
ACINFO(2) = FH between engine phase inspections
ACINFO(3) = hours to paint

```
Ex:  DATA ACINFO/1200,900,96/
```

TASKIN is a Nx5 array, where N is the number of different system level tasks (currently, there are 11 system level tasks. For each task the input includes 1) what resource is needed, 2) the number of that resource needed, 3) second resource is needed, 4) the number of second resource, and 5) the task time. If only one resource type is needed the second resource type = 0. The model recodes that resource to a dummy resource, which is always available.

TASKIN(1,J) = engine tear down, build-up
TASKIN(2,J) = autoecs
TASKIN(3,J) = test cell
TASKIN(4,J) = major airframe phase
TASKIN(5,J) = minor airframe phase
TASKIN(6,J) = tow before maintenance
TASKIN(7,J) = preflight inspection
TASKIN(8,J) = thru flight servicing
TASKIN(9,J) = postflight inspection (BPO)
TASKIN(10,J)= monthly servicing
TASKIN(11,J)= 200 FH servicing
TASKIN(12,J)= tow after maintenance

```
Ex:  DATA (TASKIN(2,J),J=1,5)/13,1,8,1,14.5/
```

WUC Level Inputs: The WUC level inputs are divided into four parts and input by array - CODES, JRSCTN, TIMES, and SHOP.

The CODES array is a Nx9 array, mostly containing probabilities of various tasks occurring for each WUC.

CODES(N,1) = MTBM for the nth WUC
CODES(N,2) = code for whether cannibalization is permitted (1=yes,0=no)
CODES(N,3) = % of NMCS aircraft (normally use total system value)
CODES(N,4) = % repairs which are on equipment
CODES(N,5) = % repairs needing a functional check
CODES(N,6) = % repairs needing a functional check flight
CODES(N,7) = % repairs resulting in an on equipment CND action
CODES(N,8) = % rejected after functional check
CODES(N,9) = % requiring towing

Ex: DATA (CODES(11,J),J=1,9)/92,1,.05,.3,1,0,.15,.05,.55/

The JRSCTN is an Nx12 array, containing the resource types and number needed by task, for each WUC.

JRSCTN(N,1) = first resource type for troubleshoot
JRSCTN(N,2) = # of first resource type for troubleshoot
JRSCTN(N,3) = second resource type for troubleshoot
JRSCTN(N,4) = # of second resource type for troubleshoot
JRSCTN(N,5) = first resource type for repair or remove/replace
JRSCTN(N,6) = # of first resource type for repair or r/r
JRSCTN(N,7) = second resource type for repair or r/r
JRSCTN(N,8) = # of second resource type for repair or r/r
JRSCTN(N,9) = first resource type for functional check
JRSCTN(N,10)= # of 1st resource type for functional check
JRSCTN(N,11)= second resource type for functional check
JRSCTN(N,12)= # of 2nd resource type for functional check

For the T46 the resource types are as follows:

AFSC	RESOURCE TYPE	AFSC	RESOURCE TYPE
325x1	1	426x2	8
328x0	2	phase dock	9
423x0	3	427x5	10
423x1	4	APG	11
423x2	5	dummy	12 or 0
423x3	6	autoecs	13
423x4	7		

Ex: JRSCTN(12,J),J=1,12/11,2,7,1,11,2,7,1,11,2,0,0/

In this example 2 APG's and 1 423x4 are needed to troubleshoot, the same people are needed for repair, and only the APG's are needed for the functional check.

The TIMES array is an Nx12 array, containing the minimum, most likely, and maximum times for each task.

TIMES(N,1) = minimum time to troubleshoot
TIMES(N,2) = most likely time to troubleshoot
TIMES(N,3) = maximum time to troubleshoot
TIMES(N,4) = minimum time for on equipment repair
TIMES(N,5) = most likely time for on equipment repair
TIMES(N,6) = maximum time for on equipment repair
TIMES(N,7) = minimum time to remove and replace
TIMES(N,8) = most likely time to remove and replace
TIMES(N,9) = maximum time to remove and replace
TIMES(N,10)= minimum time to perform functional check
TIMES(N,11)= most likely time to perform functional check
TIMES(N,12)= maximum time to perform functional check

Ex: TIMES(15,J),J=1,12/15,30,60,0,0,0,60,90,200,15,30,60/

In this example it takes between 15 and 60 minutes to troubleshoot; there is no on equipment repair; it takes between 60 and 200 minutes to remove and replace; and between 15 and 60 minutes for the functional check.

The SHOP array is an Nx5 array, containing information on the shop repair. (After a remove and replace action, the item is sent to the shop).

SHOP(N,1) = resource type
SHOP(N,2) = number of resources needed
SHOP(N,3) = minimum time for shop repair
SHOP(N,4) = most likely time for shop repair
SHOP(N,5) = maximum time for shop repair

Ex: SHOP(13,J),J=1,5/7,1,106,125,185/

SLAM Inputs: In addition to the inputs in the data file, the overall MTBM can be changed in the SLAM file. This will override the values in the data file, and will adjust each WUC MTBM by the same factor.

IV. NUMBER OF RUNS/LENGTH OF RUNS:

Any analysis should consist of a run of 3000 hours which includes a warm-up period of 500 hours.

The length of the runs is set in the INIT statement at the bottom of the file; the statement "MONTR,CLEAR,500.0;" clears all statistics at time 500.0 (warm-up).

V. TO RUN THE MODEL:

On the IBM 4341 with the CMS operating system, the mechanics of running the model are controlled by the T46A EXEC file (refer to Appendix E. Computer Code). The exec file compiles both the source code (T46A FORTRAN file) and the data (T46AD FORTRAN file), loads these files and SLAM (SLAM EXEC file) runs the program. The exec file can also be used to clean up the files by erasing the TEXT and LISTING files which are created when T46A FORTRAN and T46AD FORTRAN are compiled. These optional statements have an asterik in front of them in Appendix E. Do not use these if the model is not working, since the LISTING files may be needed for debugging.

To run the model, simply type "run T46A". Once loaded, the model takes about 10 - 15 CPU seconds for each week simulated (based on MTBM=2.9; the lower the MTBM, the longer it takes to run because more tasks are being simulated). There is a statement in the CALNDR subroutine which will print the week number to the screen at the end of each week.

VI. VERSION OF SLAM:

This model should run on all versions of SLAM later than 2.0 with one adjustment to a common block which was taken from the SLAM FORTRAN file. If the common block in the SLAM FORTRAN is not the same as listed below, change the common block in the T46A FORTRAN file to match the SLAM FORTRAN. If you do not have access to the source code for the SLAM FORTRAN, contact Pritsker & Assoc to determine the exact coding of this common block.

```
*** COMMON BLOCK GCOM1 IS USED TO SCHEDULE THE OUTPUT AT TIME TTFIN
      PARAMETER (MMXXV=100)
      COMMON/GCOM1/ JJCDR,KKNN,LLFIL,LLRNK,LLTRY,MFEX,NNAM1,NNAM2,NNAM3,
     1NNAPO,NNAPT,NNATR,NNFIL,NNTRY,TTBEG,TTCLR,TTFIN,
     2TTSET,XXI(MMXXV),TTTS,TTTF
```

The only purpose of this common block is to determine the end of the simulation (TTFIN) and schedule the subroutine STATS to occur after the simulation is complete. It is possible to delete the common block and parameter statement, and schedule event 17 at the same time as input in the T46A SLAM file. However, it's so easy to forget to change it in both places that it's worth the extra effort to change the common block.

BIBLIOGRAPHY

1. Air Force Operational Test and Evaluation Center (AFOTEC). T-46A Next-Generation Trainer Initial Operational Test and Evaluation Plan. Kirtland AFB NM, April 1985.

2. Banks, Jerry and John S. Carson, II. Discrete-Event System Simulation. New Jersey: Prentice-Hall, Inc., 1984.

3. Department of the Air Force. Air Force Reliability and Maintainability Program. AFR 800-18. Washington: HQ USAF, 15 June 1982.

4. Pritsker, A. Alan B. Introduction to Simulation and SLAM II. New York: John Wiley & Sons, Inc., 1984.

5. Drake, William F., Chief, LCOM Systems Branch, Personal Interview, AFMSMMET/MEMC, Wright Patterson AFB, OH, 6 December 1985.