A MONTE CARLO PROCESS FOR DETERMINING RESPONSE TIMES FOR TACTICAL SYSTEMS

Leroy C. Sanders

Hughes Aircraft Company
Ground Systems Group
Fullerton, California

Hughes Document Control Number TP 68-16-30

SUMMARY

The MACK tactical system is a highly Mobile Command and Control system. To monitor system performance for the MACK tactical system, analysis was conducted to support the design and development of the logic for the functional elements of the Operational and Recording Programs and their interface capability with the Executive and with associated peripheral equipments. This analysis provided the capability for continuous monitoring of the operating environment from the point of sequencing computer subprograms, allocating the necessary storage requirements and satisfying response time requirements. This analysis was conducted by the development of a Monte Carlo Process for Determining Response Times for Tactical Systems simulation program.

INTRODUCTION

The continuous demand that computerized decision making techniques in tactical systems permit rapid responses to urgent and varied externally initiated requests play a vital role in determining system performance. Clearly, the performance of an existing system can be determined from an observe-record-analyze approach. However, for systems still in their design stages, the problem of ascertaining system performance is acute. No system exists for observational purpose. Yet, some simplified method, depicting system performance, must exist in order to continually evaluate proposed system improvements. Obviously, each system's performance depends upon the system's hardware and software packages. In many instances realistic assumptions can be made about these packages so that the effect of one upon system performance can be relegated while system performance is considered to be primarily dependent upon the other.

In determining system performance for the MACK tactical system, system analysis was conducted to support the development of its hardware and software packages. However, it soon became apparent that a mathematical model was needed to aid in evaluating critical areas of the system. It was decided to simulate the MACK tactical system using the General Purpose System Simulator (GPSS) III as the technique of Analysis.

APPROACH

The approach selected for constructing the model was as follows:

I. Each equipment and program function normally.
II. The interaction is primarily among operator, external world, and software package.
III. Construction of an Executive Program which supervises and designates authority to subordinate programs.
IV. Analyze response times versus requests generated for a selected operations center.
V. Presents hard copies (and possibly plots) of significant results to management to enable an analyst to evaluate system performance.

DATA SOURCES

The processing times for the operational programs are obtained from the program development activities. These times are submitted bi-monthly and progress in accuracy from rough initial estimates to actual measurements. The auxiliary input data is determined by the user of the program.

DATA SAMPLE

The input data of interest to this study include, but is not limited to, the following:

I. Environmental Data
  A. Identification Friend or Foe/Selective Identification Feature (IFF/SIF) Data
  B. Adaptation Data
  C. Operator Actions
  D. Automatic Data Link Data
  E. Aircraft Status Data

II. Timing algorithms which define the processing times for the various operational and recording programs.

OBJECTIVES

The program was written to extract the following information:

I. Operational and recording programs processing times versus system load.
II. The effect of interrupts on response times.
III. Elapsed times required to insert (retrieve) data into (from) the computer.
IV. Core estimates of selected memory banks.

THE SIMULATION

Data Initiation and Data Flow

Simulation studies are limited only by our imagination and available computer core storage. Thus, it is possible to derive and program a mathematical model which represents, approximately, the system which is being considered. In this study the assumption was to imagine that an operations center exists; modularly constructed. System inputs result from various actions being generated at any of fourteen consoles, at the operator station, Automatic Data Link (ADL) Network or other equipments. Figures 1, 2 and 3 depict the peripheral equipments attached to the computer, and the flow of information through the system. Finally, the need exists to simulate the system for any operational period, repeat the process as often as desired in a single "run", but restrict the computer time to a few minutes.

To properly simulate an action generated at a console, operator station, or other equipments requires that data be selected at random or in an orderly manner. To ensure that both capabilities exist, a Monte Carlo process is incorporated. In this case the random variable may be either the type of action generated or the time at which the action is generated or both. The action taken at a console, operator station, or other equipments is selected from one of several stored actions for that console, operator station or other equipments. The stored actions are defined as FUNCTION$_{al}$ values.

The simulation program permits multiple entries of input data to be stored internally before a complete message is created. The program also allows the hardware to process inputs as a function of the number of bits associated with the message. Thus, as an input moves from its point of initiation to its first core location, one of its delay times can be computed as a function of the number of bits associated with the message.

In the model, the time of occurrence of each initial action is input via an INITIAL card. After the initial input is generated, all succeeding inputs initiated at this source are generated via one of two methods (depending upon program inputs). In one method the succeeding action or time or both action and time is (are) randomly selected. In the second method each source generates actions cyclically at unique constant time intervals.

If the generated action originates from a console, the message is delayed in the Control and Coding Logic and Output Data Unit blocks (see FIGURE 2) before it is QUEUE$_d$. The message will remain in the appropriate QUEUE until one of two conditions are satisfied: (1) a K-Millisecond interrupt occurs and the hardware initiates its display cycle; (2) a hardware display cycle is terminated after more than K-Milliseconds have elapsed. In either case, the appropriate QUEUE$_s$ are interrogated (by the hardware package) for their inputs prior to the initiation of each display cycle. Only one input from each QUEUE is allowed to leave at the time of interrogation. The departing input enters sequentially the Data Converter and the Console Buffer. The Computer Input/Output (Computer I/O) processor takes the input from the buffer and stores it into its initial core storage. The input is then under the influence of the Executive program. In case the input is initiated from the operator station or magnetic tape synchronizer, the input is delayed in the peripheral I/O area before it is QUEUE$_d$ (see FIGURE 3). It will remain in the QUEUE until the peripheral I/O program (of the Time Cycle) is activated. When this program is activated, all inputs currently residing in the peripheral I/O inputs QUEUE$_s$ are removed from the QUEUE$_s$ via Computer I/O processor and stored in core memory. Again, each input is now under the influence of the Executive program.

When IFF/SIF data is required, one IFF/SIF input is generated. Instead of defining a FUNCTION for the type of desired IFF/SIF inputs, the FUNCTION$_{al}$ values represent the added number, N, of IFF/SIF inputs created this period. The parent IFF/SIF input is duplicated N times. The Computer I/O accesses the (N+1) inputs and stores them in the IFF/SIF buffer. The IFF/SIF inputs are then under the influence of the Executive program.

If the generated action represents an ADL input, the FUNCTION$_{al}$ values of thirteen unique FUNCTION$_s$ are stored into the first thirteen PARAMETER$_s$ associated with this input. Each PARAMETER represents a unique ADL message type and its value is the magnitude of that message type. A similar approach is utilized in initiating ADL outputs. If the message is an ADL input, the message is delayed to represent a transit delay between tracking sites. Both ADL input and output messages are stored in unique storages. The messages are then under the influence of the Executive program.

An appropriate time delay is allowed to elapse each time the Computer I/O processor accesses and stores each input into core. The Computer I/O processor processes each input on a priority basis.

Executive

The Executive Program consists of several subprograms. The list includes: Interrupt Receiver, Time Cycle Controller, Intermediate Cycle Controller and the Data Cycle Controller. The Executive supervises the activities of the operational and recording programs. It segregates the operational and recording programs into three classes (commonly called cycles): the Time Cycle (see FIGURE 5) for quick response, rapid processing programs; the Intermediate Cycle (see FIGURE 6) for quick response but less rapid processing programs; and the Data Cycle (see FIGURE 7) programs which are essentially slow response and slow processing programs.

Interrupt Receiver. This subprogram inhibits further interrupts while an existing interrupt is being serviced. During lockout, interrupts are stacked and then processed according to priority. The types of interrupts processed by the Interrupt Receiver include: Display Output Interrupt, North Pulse Interrupt and the K-Milliseconds interrupt which initiates the Time Cycle. The Display Output Interrupt occurs at the end of each display cycle. The North pulse interrupt occurs upon detection of North pulse.

Time Cycle Controller. This subprogram is entered at the occurrence of the K-Milliseconds Interrupt. The Time Cycle Controller sequences through the list of Time Cycle programs, executing each program in turn if frequency counts and need indicators are set. Each execution causes data to be transferred between STORAGEs. When the list is completed control is transferred to the Intermediate Cycle Controller unless the last K-Milliseconds Interrupt occured during the Intermediate Cycle. In the latter case, the previous machine conditions are restored and control is returned to the point of interrupt. Before entering the Intermediate Cycle the transit time through the Time Cycle is recorded. Time spent out of the Time Cycle (to process superior interrupts) is also recorded.

Intermediate Cycle Controller. This program steps through the list of Intermediate Cycle programs, executing each program if frequency counts and need indicators are satisfied. Each execution causes data to be transferred between $STORAGE_S$. When the list is completed the transit time through the Intermediate Cycle is recorded. Time spent out of the Intermediate Cycle (to process superior interrupts) is also recorded. Previous machine conditions are restored and control is returned to the point of interrupt in the Data Cycle.

Data Cycle Controller. This program steps through a circular list of Data Cycle programs, executing each program if frequency counts and need indicators are satisfied. Each execution causes data to be transferred between $STORAGE_S$. When the list is completed the transit time through the Data Cycle is completed. Time spent out of the Data Cycle (to process superior interrups) is also recorded.

Time-Intermediate-Data Cycle Programs

The 1000 maximum words count imposed upon this paper does not permit the author to describe the individual programs which make up the Time, Intermediate, and Data Cycles. The primary function of these programs is to move the data, created during "Data Initiation and Data Flow", through various "states" to represent system processing. An appropriate time delay is allowed to elapse whenever program processing is required. For the MACK tactical system each program performs a definite function. However, the various programs depicted in Figures 5, 6 and 7 are merely dummy programs. This implies that the model can be used to simulate an arbitrary tactical system by assigning specific tasks to the different programs.

CONCLUSIONS

The results from this simulation study clearly demonstrated that the MACK tactical system was capable of meeting response times under adverse system loads. Moreover, the results showed that predicted QUEUE lengths agreed with computed QUEUE lengths. A measure of the likelihood of being in any cycle was computed. The likelihood function provided a quick and easy way of determining the amount of time the Executive allocated to each cycle. The results from the model and the increased interest in the model may lead management to develop a more sophisticated model to aid in the design and planning stages of future tactical systems.

ACKNOWLEDGEMENT

My work is supported by the United States Air Force, Contract F19628-67-C-0154. However, the content of this paper does not imply approval by the Air Force.

82632-1

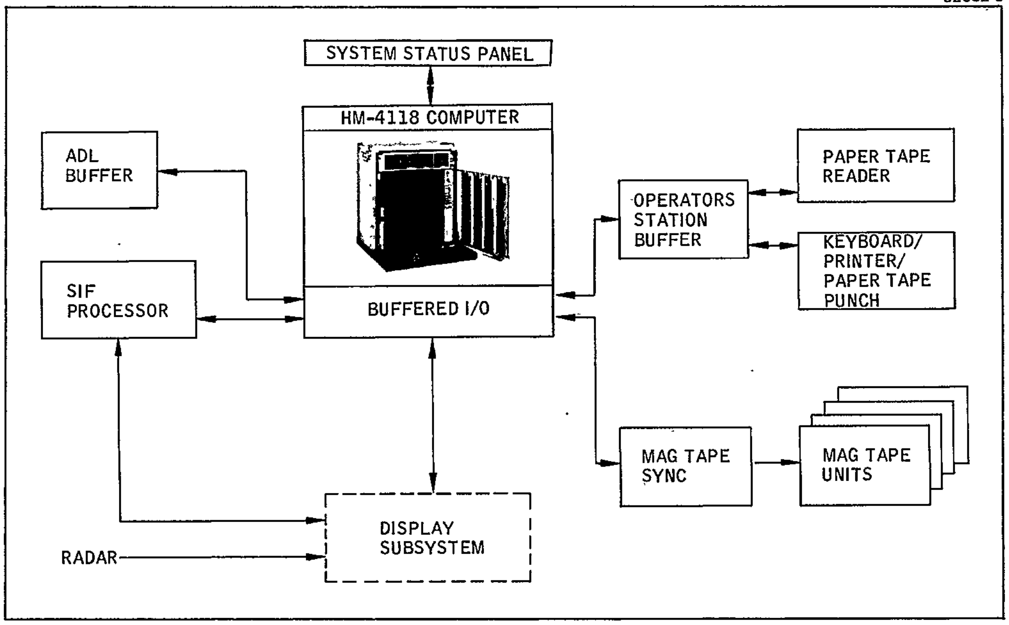

Figure 1. MACK Operations Center CRC/CRP Design — Data Processing Subsystem

82632-2

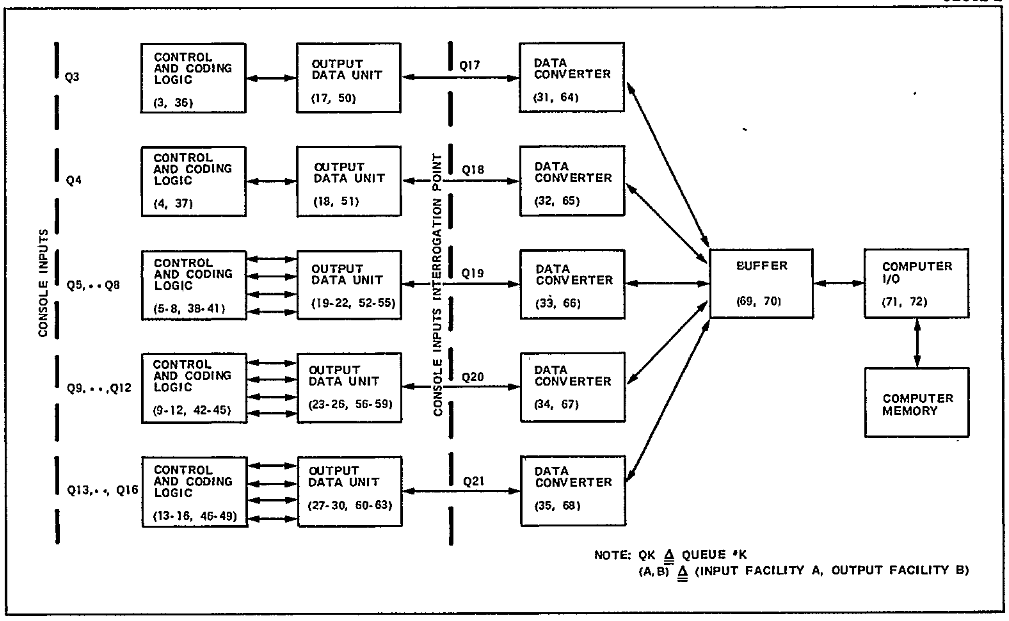
CONSOLE INPUTS
Q3
Q4
Q5, • • Q8
Q9, • • •, Q12
Q13, • •, Q16
CONTROL AND CODING LOGIC (3, 36)
CONTROL AND CODING LOGIC (4, 37)
CONTROL AND CODING LOGIC (5-8, 38-41)
CONTROL AND CODING LOGIC (9-12, 42-45)
CONTROL AND CODING LOGIC (13-16, 46-49)
OUTPUT DATA UNIT (17, 50)
OUTPUT DATA UNIT (18, 51)
OUTPUT DATA UNIT (19-22, 52-55)
OUTPUT DATA UNIT (23-26, 56-59)
OUTPUT DATA UNIT (27-30, 60-63)
CONSOLE INPUTS INTERROGATION POINT
Q17
Q18
Q19
Q20
Q21
DATA CONVERTER (31, 64)
DATA CONVERTER (32, 65)
DATA CONVERTER (33, 66)
DATA CONVERTER (34, 67)
DATA CONVERTER (35, 68)
BUFFER (69, 70)
COMPUTER I/O (71, 72)
COMPUTER MEMORY
NOTE: QK ≜ QUEUE #K
(A,B) ≜ (INPUT FACILITY A, OUTPUT FACILITY B)

Figure 2. Console I/O Configuration

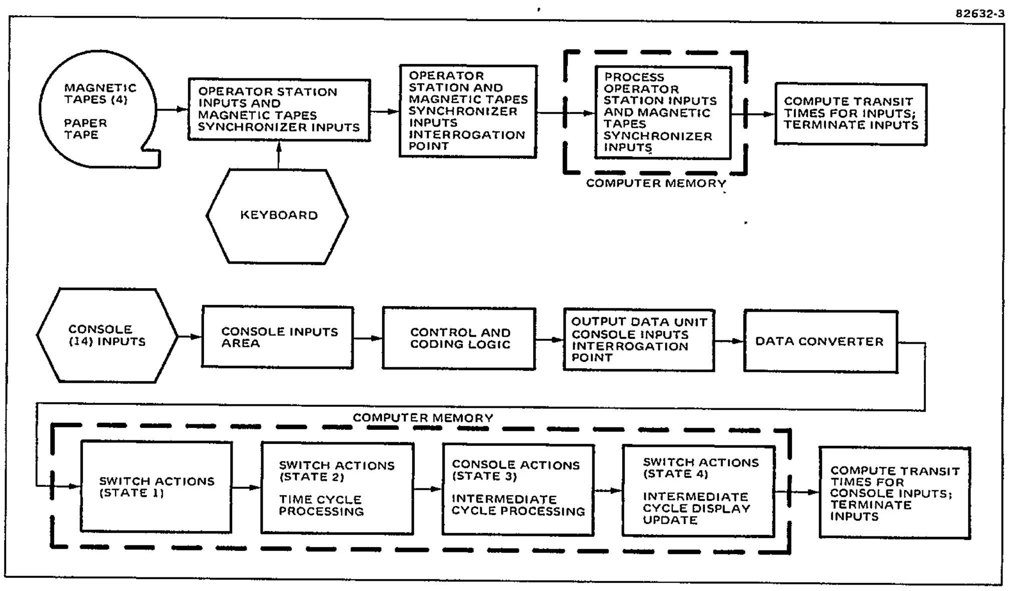
82632-3
MAGNETIC TAPES (4)
PAPER TAPE
OPERATOR STATION INPUTS AND MAGNETIC TAPES SYNCHRONIZER INPUTS
KEYBOARD
OPERATOR STATION AND MAGNETIC TAPES SYNCHRONIZER INPUTS INTERROGATION POINT
PROCESS OPERATOR STATION INPUTS AND MAGNETIC TAPES SYNCHRONIZER INPUTS
COMPUTER MEMORY
COMPUTE TRANSIT TIMES FOR INPUTS; TERMINATE INPUTS
CONSOLE (14) INPUTS
CONSOLE INPUTS AREA
CONTROL AND CODING LOGIC
OUTPUT DATA UNIT CONSOLE INPUTS INTERROGATION POINT
DATA CONVERTER
COMPUTER MEMORY
SWITCH ACTIONS (STATE 1)
SWITCH ACTIONS (STATE 2) TIME CYCLE PROCESSING
CONSOLE ACTIONS (STATE 3) INTERMEDIATE CYCLE PROCESSING
SWITCH ACTIONS (STATE 4) INTERMEDIATE CYCLE DISPLAY UPDATE
COMPUTE TRANSIT TIMES FOR CONSOLE INPUTS; TERMINATE INPUTS

Figure 3. Data Initiation and Data Flow

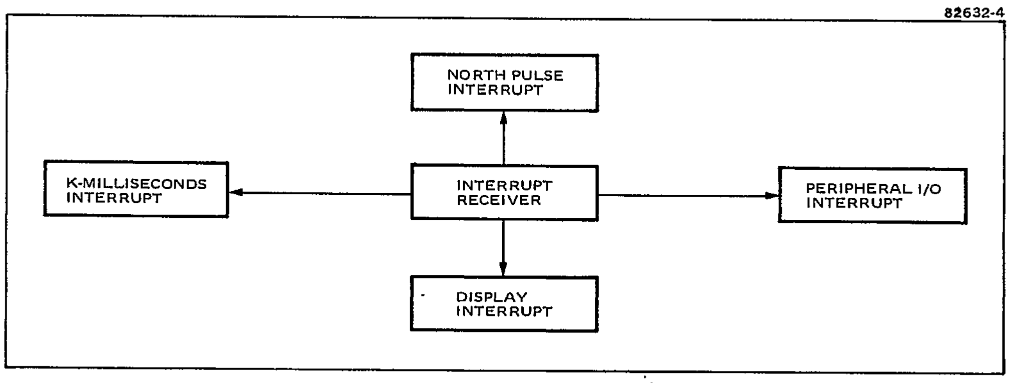

Figure 4. Interrupt Philosophy

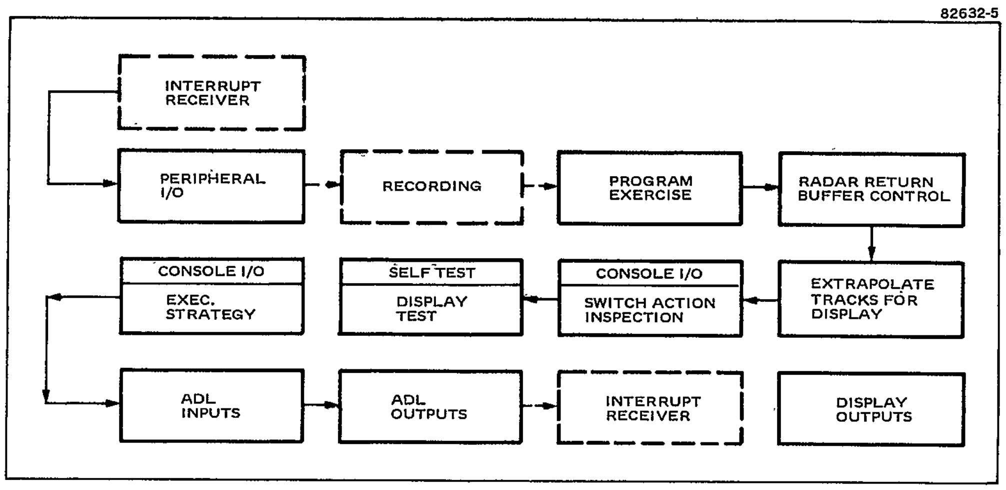

Figure 5. Time Cycle Programs and Displays

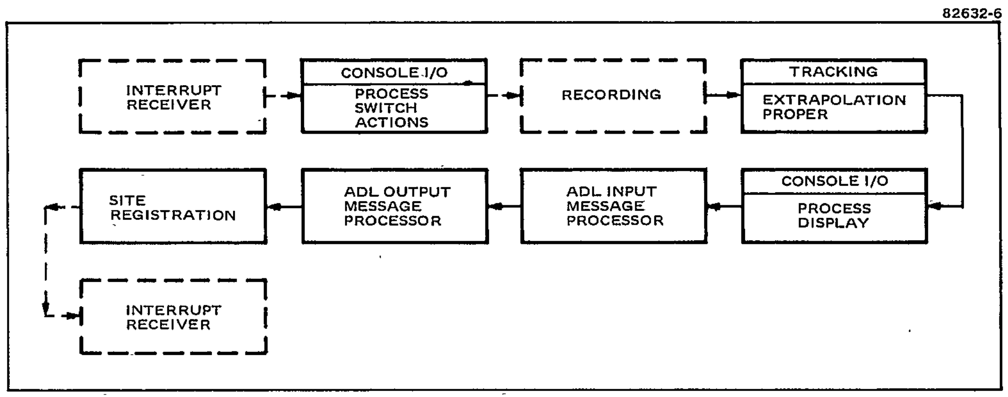

Figure 6. Intermediate Cycle Programs

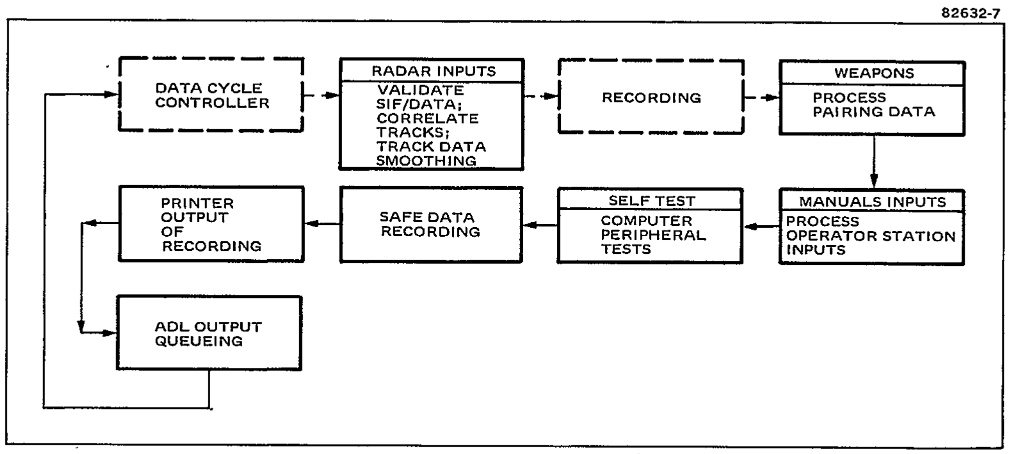
82632-7
DATA CYCLE CONTROLLER
RADAR INPUTS
VALIDATE SIF/DATA; CORRELATE TRACKS; TRACK DATA SMOOTHING
RECORDING
WEAPONS
PROCESS PAIRING DATA
PRINTER OUTPUT OF RECORDING
SAFE DATA RECORDING
SELF TEST
COMPUTER PERIPHERAL TESTS
MANUALS INPUTS
PROCESS OPERATOR STATION INPUTS
ADL OUTPUT QUEUEING

Figure 7. Data Cycle Programs