4

ARL-STRUC-TM-488

AR-005-525

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Aircraft Structures Technical Memorandum 488

SINGLE CHANNEL TEST CONTROLLERS (U)

by

I. Powlesland and E.S. Moody

Approved for Public Release

JUNE 1988

AR-005-525

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Aircraft Structures Technical Memorandum 488

SINGLE CHANNEL TEST CONTROLLERS (U)

by

I. Powlesland and E.S. Moody

SUMMARY

The stand-alone controllers in single and coupled-channel forms are described. Details of hardware and software are given, and notes on the associated data acquisition system and test boxes are included. A number of appendices provide data for preparation and reading of floppy disks, for cycling and communication rate adjustments and detailed information on other aspects of these units.

DSTO ARL
MELBOURNE

POSTAL ADDRESS: Director, Aeronautical Research Laboratory,
P.O. Box 4331, Melbourne, Victoria, 3001, Australia

CONTENTS

		Page No
1.	INTRODUCTION	1
2.	SYSTEM DESIGN OBJECTIVES	1
3.	SYSTEM HARDWARE	2
4.	SYSTEM SOFTWARE	7
5.	COUPLED SYSTEMS	17
6.	DATA ACQUISITION SYSTEM	19
7.	CONCLUSION	23
8.	REFERENCES	23

FIGURES

DISTRIBUTION LIST

DOCUMENT CONTROL DATA

1. INTRODUCTION

Electro-hydraulic control of the test loads applied to major aircraft structural components was first used at the Aeronautical Research Laboratories in 1971. While the basic feedback control loop used at that time has been changed only in matters of detail there have been significant changes in the loading profiles required. Changes have also occurred in the techniques employed to generate loading waveforms and to set and monitor the applied load values.

In recent times most of the development effort has been directed to the large computer-controlled multi-actuator systems used in the fatigue testing of full scale aircraft. The original single channel system has continued in service until failing reliability and non-availability of parts made replacement essential. This single channel system provided constant amplitude sinusoidal loading with amplitudes changing by pre-set amounts after completion of a specified number of cycles at each amplitude. This loading profile was generally referred to as 'Block Programming'. An essential requirement of any new single channel system would be the provision of 'Flight-by-Flight' load profiles, wherein load amplitudes are distributed in an irregular pattern, on a single cycle basis, to simulate the loading occurring in flight. Other facilities provided in the multi-actuator systems, and considered desirable for single channel systems, include 'Load value at turning point' monitoring, strain measurements at turning points and automatic strain-versus-load calculations of the test specimen.

The system described in this report brings all of these features and more to single channel structural component testing.

2. SYSTEM DESIGN OBJECTIVES

2.1 Specification

The broad specification directing the design effort was expressed in four major objectives as follows:

(1) To provide a single channel electro-hydraulic feedback controller and load programmer requiring minimum operator supervision. This self-contained, self-checking and self-protecting system should provide a hard copy log of test progress.

(2) To provide sensors for specimen monitoring at each specified load value of a test program. The number of sensors could be limited but sensor amplifiers should be mounted as close to the sensors as possible.

(3) To provide all the features of the original controller and those features of the advanced multi-channel systems considered relevant to single channel applications. The controller should be capable of both 'block' and 'flight-by-flight' programming, and load and strain monitoring should be provided.

(4) To provide a system suitable for the study of distributed controllers wherein a number of single channel controllers may work together in synchronism, either without external supervision or under the control of a mini-computer.

2.2 The Self Checking Features

2.2.1 Load range

The amount of load to be applied by a full scale input command to a feedback load controller is determined by the gain of the feedback amplifier. It is essential this gain be correctly set for the desired load range.

The new single channel controllers were required to include facilities for automatic verification of feedback amplifier gain. This was to be achieved by the automatic application of predetermined command and shunt-stimulated load cell signals appropriate to the required load range. When the amplifier gain was correctly set these signals would sum to zero.

2.2.2 Load turning point values

In multi-channel systems automatic tests were made at each load turning point to establish command/load agreement. These tests required digitisation of independent load measurements and comparison of digital values of command and load.

Similar verification of load turning point values was required in the single channel controllers. For simplicity analogue comparisons of command with both feedback signal and independent load signal, using separate null detecting circuits, were specified. Simultaneous null signals from both detectors indicated satisfactory command/load agreement.

3. SYSTEM HARDWARE

3.1 System Configuration

The system comprised four 3E high 19 inch (483 mm) modules, an input termination box and up to eight sensor amplifiers, Figure 1. Load data was presented to the system on an eight inch (203 mm) double sided, double density floppy disk read by the Load Program Module. The analogue command generated by this module was output to a Servo Amplifier Mark 2A in the Rig Monitor Module. This module monitored load cell output as well as providing servo valve control.

The DC power supply for the Load Program Module and the Rig Monitor Module was located in the Hydraulic Control Module. This module controlled oil flow to the rig by providing electrical signals to operate the solenoid valves in the hydraulic circuit.

The Data Acquisition Module accepted signals from up to eight sensors. More than one data acquisition module could be used if required. Output signals amplified adjacent to the sensors were digitised in this module and recorded on an internal eight inch (203 mm) double sided, double density floppy disk. For more details of this subsystem refer to Section 6.

3.2 Hardware Selection

Some off the reasons for the section of particular system components are discussed in the following paragraphs. Non-technical considerations governed the selection of some components.

3.2.1 Memory

Memory components performed four functions in this system.

(1) Bulk storage of load program data and specimen response data. The eight inch (203 mm) floppy disks were selected for both applications because they provided compact data storage on a readily transportable medium. Their performance under fatigue test conditions remained to be demonstrated but they were used extensively throughout the computer industry. The storage capacity provided by these double density disks was adequate for load program storage but somewhat limited for continuous sensor data recording.

At the time the selection was made Winchester disk technology was available but relatively expensive and lacked the simple controller hardware available for floppy disks. Interchange of disks was more readily achieved with floppy disks than with Winchester cartridges. Small cassette tape units lacked the necessary storage capacity and did not provide the rapid restart and repeat facilities considered necessary. Generally these units were not available in the simple, robust and inexpensive formats required.

Bubble memory had been used in another application, but with a very uncertain supplier situation the future of these units looked bleak. Availability improved recently but their capacity remained only marginally adequate. Portability of load and sensor data records would be less readily achieved than with floppy disks. However if floppy disks failed to provide adequate performance in this application bubble memories would now be an alternative worthy of consideration.

(2) Permanent storage of test status to allow restart after accidental shutdown. To avoid special, time critical shutdown routines the memory used for test status storage was required to be part of the working memory of the system. CMOS random access memory was selected for the purpose because it was fully compatible with the rest of system RAM. Its very low power requirement meant it could be maintained for long periods by a small, low capacity battery. With this CMOS memory used directly as workspace registers there was no requirement for special routines to save data during either controlled or accidental shutdown.

Electrically Erasable Programmable Read Only Memory (EEPROM) was considered as a possible alternative to the CMOS memory. It was rejected because its relatively slow speed would prohibit use in main memory, and undesirable transfer routines would be necessary. Because this technology was then in its infancy power supply and timing requirements were both complex and critical; and local availability of devices and technical data was uncertain.

(3) Program storage. The application program and a primitive disk operating system were stored in EPROM. EPROMS were chosen in preference to the smaller and less expensive PROMS because the number of units to be produced was small, program modification was facilitated, and a suitable EPROM programmer was readily available.

(4) Working memory. Static RAM was selected for this purpose because the amount of random access memory required was small, and the added complexity of dynamic RAM was not justified.

3.2.2 The microprocessor

The TMS9900 was selected for this application because considerable experience in the development of hardware and software for it had been accumulated, and essential tools for program development were available. This unit had a full 16-bit data bus and provided hardware multiply and divide capability. Its address modes, communication register unit and operating speed were particularly suitable for this application; and members of its family of very advanced support chips were suitable for use in the proposed system.

3.2.3 Operator-system interface

A unit was required which would permit interaction of operator and controller and would provide a hard copy test log. The Teletype Model 43 was chosen because it had a proven performance record under office conditions and it was familiar to the majority of potential operators. The teletype provided a durable hard copy of adequate quality and was sufficiently flexible to meet potential system development. It was robust and reasonably portable.

3.2.4 The system cards

The familiar printed circuit and connector had been a source of trouble in several installations operating in conditions better than those to be experienced by these controllers. Consequently this configuration was discarded in favour of the indirect, card-mounted connector. This connector had the added advantage that connector density could be increased considerably without increasing the complexity of manufacture of the card. The mechanical arrangement of Cards in this system generally followed that recognised as the 'Eurocard' configuration.

3.3 Load Program Module

This module contained a disk drive and five major circuit card assemblies.

3.3.1 Mechanical arrangement

Most of the Cards were rear loaded into the 3E x 89T x 4248 mm card frame and mated with an insulation displacement ribbon cable backplane. The exception to this arrangement was the CPU card which was constructed to an existing design and positioned above the disk drive. For the limited production run anticipated, this arrangement appeared to offer the lowest overall cost.

3.3.2 CPU card

This card contained the TMS9900, up to 16K of EPROM, 4K of RAM, a diagnostic teletype port and support circuitry. Memory mapping on this card used a PROM (74S288, DM8578, 82S123 or equivalent) which had to be replaced if different sized EPROMS (2708,2716,2732) were installed. Refer Appendix 1 and 2.

Because the teletype, interrupts and clocks were all controlled through the communications register unit (CRU), a facility not available on most other processors, it would be difficult to change to another processor type. The DMA relied upon the processor relinquishing the bus for one cycle every 11.5 microseconds, another feature also not generally available.

3.3.3 Disk controller card

This card was designed around the Western Digital 1797 controller chip, with phase-locked loop data separation and DMA data transfer. The card was configured for double density, 256 bytes per sector format. With 26 sectors per track and 77 tracks per side this yielded a total capacity of about one million bytes. The Western Digital 1797 was chosen because it was widely used and readily available.

When formatting a disk on this machine a lead from the test point on card 'A' to pin 5 on the rear CPU connector was required. This enabled the program to synchronise with the DMA system without the overhead of another counter. This additional overhead would have made it difficult to get the program to run fast enough.

Using the DMA controller, data was transferred from a specific sector on the load disk, via the disk controller, to the DMA latch one byte at a time. Transfer from the DMA latch to memory occurred one 16-bit word at a time. As each byte was received and assembled by the 1797 (at intervals of about 16 microseconds) the controller examined MEMEN for availability of the memory bus. When the bus became available (in less than 2 microseconds) a word transfer to memory took place.

The DMA latch had 16 bit capacity organised as two one-byte latches, and incoming data bytes were moved alternately to one or the other of these latches. The DMA address counter and the latch controls were so arranged that as soon as a byte of data had been accepted by the latch the current 16 bit words of data were transferred only when the byte count and address count were even. When the logical byte count was odd only one byte of the transferred word was valid. The other byte was either rubbish or carry over from the previous word. The words transferred on odd counts were eliminated from memory by being overwritten when the even count words were transferred (NB: As the physical counter started from 0, 'odd' transfer occurred when it showed even number, and vice versa).

3.3.4 Teletype card

The calendar clock (MSM5832), interrupt controller (TMS9901) and two asynchronous communication controllers (TMS9902) were mounted on this card. Three nickel-cadimium button cells mounted on the card provided power to the clock for a period of up to three months without re-charging, so that the clock settings were maintained even when the card was removed from the system. These cells were automatically charged whenever mains power was applied to the system.

Only 5 levels of interrupt were implemented in this system:

Level 0	reserved for the reset function
Level 1	used to signal a safety circuit latch trip
Level 3	used to signal to end of a timer period
Level 5	used by the asynchronous communication controller serving the TTY port
Level 7	used by the asynchronous communication controller serving the AUX port

3.3.5 Rig interface card

The major component on this card was a DAC80 digital-to-analogue converter used to provide the analogue voltage commands for the servo amplifiers. The DAC80 had industry standard pin-outs and was available from a number of manufacturers. To permit operation from 12 volt supply rails the DAC was configured for ± 5 volt output for full scale input.

The communication register unit (CRU) of the control processor was connected to the external system by circuits on this card. Single bit signals were used to control some system functions and to monitor their condition. These signals either arose in the CRU or were returned to it for tests.

A currently unused connector SK.4 mounted on this card provided direct access to the 2's complement digital command signal applied to the input of the DAC80.

3.3.6 Memory card

This card carried the battery backed CMOS memory chips used to ensure retention of test status in the event of system power failure. The memory chips were 74C921/NMC6552, selected for low standby current and ready availability. When mains power was removed the memory was maintained by a battery of three B150T cells (as used on the teletype card). This battery enabled the 0.5K bytes of CMOS memory to provide reliable storage of data for considerable periods of time.

The remaining circuitry on this card, in addition to preventing loss of stored data during power up/down cycles also provided address decoding, data buffering and timers for CMOS memory control.

3.4 Rig Monitor Module

This module contained the bulk of the system's analogue circuitry. It was constructed around the existing Mark 2A servo amplifier. Refer to Ref. 2 and drawing PL55474. The major functional areas of the module other than the servo amplifier were as set out in the following paragraphs. To simplify the manufacturing these functions were implemented on a single, large horizontal printed circuit board.

(1) Bridge Excitation. Independent high stability, preset constant current supplies were used to energise the two strain gauge bridges on the system load cell.

(2) Low Level Amplifiers. Low drift, high gain instrumentation amplifiers were used to raise the level of strain gauge bridge signals to that of the input commands.

(3) Command Amplifier. This amplifier was used to raise the ±5 volt DAC output command signal to the ±10 volt input signal range of the servo amplifier.

(4) Null Detector. When both amplified feedback and monitor strain gauge bridge signals were within preset bands about the command signal value corresponding null signals were produced. A composite null signal was passed to the Load Program Module only when both these null signals were present.

(5) Bridge Integrity Check. Relay circuitry placed pre-selected shunt 'Calibration' registers across one arm of each of the strain gauge bridges. This check permitted verification of correct operation of the above components, the DAC and the input cabling.

(6) Local Monitoring. One of a number of variables could be manually selected and displayed on a panel-mounted digital voltmeter having a one milli-volt resolution.

3.5 Hydraulic Module

When this module was designed the practice was to provide reduced pressure and/or flow at start-up, and rig lock-up on specimen failure. The module therefore made provision for the control of the following valves:

- Dump Valve - This valve allowed the return of oil through the valve pack to tank.

- Supply Valve - This valve allowed oil to reach the valve pack from the supply main.

- High Pressure Valve - This valve allowed full pressure to be applied by-passing the pressure/flow regulator.

- Actuator Release Valve - This valve allowed oil to flow between valve pack and loading actuator.

These valves were assumed to have 240 volt AC solenoids, and to minimise electrical interference, switching was by means of photo isolated, zero-crossing relays. These switching relays were controlled from an interlocking circuit with inputs from manual push buttons, limit travel micro-switches, servo amplifier error detectors, and emergency stop buttons as well as from the Load Program Module.

The circuits of this module and those of the Load Program and Rig Monitor Modules, but excluding the servo amplifier, were all powered from linear DC power supplies in the Hydraulic Module. The supplies were -12, +5, +12 and +24 volts, all with respect to one digital common, for the digital circuitry; and ±15 volts with respect to a floating analogue common for the amplifiers and other analogue circuits. The analogue and digital commons were connected together at a single point on the rig interface card in the Load Program Module.

4. SYSTEM SOFTWARE

Two distinct packages of software could be recognised; a facilities package with identical copies being located in both Load Programmer Module and Data acquisition Module, and individual applications packages.

4.1 The Facilities Package

Each of the system microcomputers had a simplified operating system TDOS4 installed in EPROM. Each of the facilities provided by TDOS4 could be used by the operator, and some could be called by the applications package. To call up a facility the operator was required to type the character-pair associated, on a teletype or similar RS-232 terminal connected to the TEST port on the front of the module and set to 4.8K baud or below. While most of the following facilities were used in setting up the system, or for testing or amendment of application programs, several were called directly by the applications program. Reading from and writing to disk were examples.

The facilities package provided 19 routines as described in the following paragraphs:

(1) Help - HE

This routine provided a printed list of all the facilities available to the operator.

(2) Read Disk - RD

This routine read a sector from the disk and transferred it to RAM starting at a given RAM address. In response to 'B' the operator typed a hexadecimal number from 0 to FA3 followed by Carriage Return, thereby specifying the sector to be read. In 77 tracks with 26 sectors in each, on both sides of the disk 4004 sectors were recorded.

In response to "P" the operator typed a hexadecimal number to specify the start address in RAM where the sector was to be loaded. This start address had to be on a 256 byte boundary i.e. hexadecimal 100,200,300 etc. It was necessary to remember that any program already present in this section of RAM would be destroyed.

After a short interval for reading and transfer of data the system printed a number from 0000 to 0003.

. If the number was 0000 the transfer had been completed satisfactorily.

. If the number was 0001 the system had been unable to run the disk. Check that the disk was properly mounted.

. If the number was 0002 a Seek Error had occurred. System Reset and retry would be required.

. If the number was 0003 the requested sector was not found.

. If System Reset and retry also failed, disk formatting errors would be suspected.

(3) Write Disk - WD

This routine wrote data to the disk sector specified by the operator in response to the prompt 'B'. The data to be transferred to disk would be taken from a block of 256 bytes of RAM starting from the hexadecimal address input by the operator in response to prompt 'P'. This command was properly executed only when any 'disk-write' protection was disabled.

The starting address in RAM had to be on an even 256 byte boundary. After a short interval the system responded with a number from 0000 to 0003 where the numbers had equivalent meanings to those given for the 'Read Disk' routine.

(4) Format Disk - FD

Great care was necessary in the use of this routine because any data on the disk at the time the routine was called would be automatically and irrecoverably destroyed. With some disks it was necessary to attach a write-enable tag before anything could be written on the disk. This routine formatted 8 inch, double sided, double density disks to give 26 sectors in each of 77 tracks on both sides of the disk. The format was IBM System 34-256 bytes per sector, described in Appendix 2.

To reduce the possibility of accidental loss of data the system asked for confirmation whenever it received 'FD' by printing 'FD?'. To continue the operator had to type 'Y', any other character would abort the routine. Several minutes were required to completely format the disk, and when the process terminated the system prompt '#' was printed.

(5) Load Disk - LD

This routine was provided for those users without compatible 8 inch floppy disk systems on their main computer. It permitted the transfer of properly formatted loads data from a central computer to a disk in the Load Program Module. The transfer took place over an RS-232 link into a 256 byte temporary buffer in the module. The central computer was required to send a sector of data in response to each ASCII 'S' from the Load Program Module.

The RS-232 line was connected to the AUX port of the Load Program Module. Both central computer and module had to be set to the same baud rate. The default rate for the module was 9.6K baud, and if the central computer required the use of some other rate reference should be made to Appendix 4.

(6) Disk Address - DA

This routine supplied the operator with the current address of the disk heads. In response to 'DA' the system printed six numbers each of two characters. First the track number starting from 00, then the side number 00 or 01, followed by the physical sector number starting from 01. This should not be confused with the block (sector) numbers given in response to 'B' in READ DISK and WRITE DISK which started from zero. The fourth number represented sector length. For 256 byte sectors the number was 01. The remaining numbers were cyclic redundancy check characters and could be ignored.

(7) Read Console - RC

There were three RS-232 ports on the Load Program Module and on the Data Acquisition Module. The TEST port mounted horizontally on the front of the module was used to call up these routines. The two remaining ports were on the back of the module and labelled TTY and AUX respectively. TTY was the Console port and AUX the Auxiliary port. When the operator input 'RC' on the TEST port terminal the system waited for a character to be input at the Console (TTY) port. Any character input on the Console was then typed on the TEST terminal.

(8) Write Console - WC

This routine was similar to 'RC' but printed on the Console terminal any character which was input on the TEST terminal.

(9) Read Auxiliary - RA

As for the 'RC' routine, the character was input at the AUX port terminal and printed on the TEST terminal.

(10) Write Auxiliary - WA

As for 'RA' routine, a character input on the TEST terminal was printed on the AUX port terminal from the TTY and AUX port terminals, Refer to Appendix 4,5.

(11) Set Bit - SB

This routine set to one or cleared to 0 any specified output bit on the Communication Register Unit (CRU). The command format was:

- 'SB', bit number (expressed in hex), space, required state of bit (0 or 1) and carriage return.

For this routine all CRU bits were numbered relative to base 0.

(12) Test Bit - TB

This routine permitted the examination of any CRU input bit. The operator typed:

- 'TB', bit number (in hex) and a space. The system typed a 1 or a 0 depending upon whether the nominated bit was set or cleared.

Bits were numbered relative to base 0 as for 'SB'.

(13) Set Clock - SC

Both microcomputers had calendar clocks which indicated date (day, month and year), and time (hours and minutes). In response to 'SC' the system typed:

DD MM YY - HH MM

The operator inserted a character under each letter using leading zeros where necessary to initialise the clock. The system repositioned the terminal carriage after each entry and the operator was not required to input any spaces or carriage returns.

Note: DD is Day, MM is Month and Minutes, YY is Year and HH is for Hours.

(14) TIME - TI

In response to 'TI' the system typed the current reading of the clock in the format given above, with leading zeros retained.

(15) Address Modifier - AM

This routine permitted the examination of any word in memory and the alteration of words in RAM. The operator typed 'AM' followed by the hexadecimal address of the first memory location to be examined, followed by a space. This was a word based routine and the hexadecimal address had to be an even number. Once 'space' was typed the system printed the contents of that memory location as four hexadecimal characters. If 'space' was again typed the contents of the next memory location would be printed and so on. If the word in any memory (RAM) location was to be changed, the typing of 'space' following the presentation of the location's current contents was postponed until the required contents had been typed. The contents were not changed until the following 'space' was typed. The word then displayed was the contents of the next memory location, not the amended location. To exit 'AM' the operator typed either Carriage Return or Break.

(16) Cycle - CY

This routine checked the operation of the Digital-to-Analogue converter of the Load Program Module and the Analogue-to-Digital converter of the Data Acquisition Module. For the DAC a waveform was generated, and for the ADC an 8-channel 'Read' was performed with the readings printed on the TEST terminal. The routine assumed both converters were present but ran for either. It was a continuous process and could be terminated only by pressing the module reset button.

(17) G0 -- GO

'Go' had to be followed by a memory address. This routine switched the program counter to that address to begin execution of program code. It was useful in program development with a TEST terminal but was not required in a working system. For example 'GO E000' would begin operation of the load programmer.

(18) Load Program - LP

This routine transferred program or data from a source computer to RAM in either the Load Program Module or the Data Acquisition Module using an RS-232 line connected to the module's TEST port. Both computer and module had to be operating at the same baud rate, and the data to be transferred had to be in the Technico Dump format. This format required the data to be preceded by a 4-ASCII-character starting address followed by a colon, followed by a 'space'. The data was required to be in pairs of ASCII characters separated by spaces with each pair representing one byte.

The source computer file to be transferred had to begin with 'LP' to instruct the module to take the data. Any characters between 'LP' and the four ASCII address characters immediately before the colon were ignored. The transfer of data continued until the source computer stopped sending, and the data was stored sequentially in the module. Once transmission of data was complete the source computer could send a 'Break' command or the operator could press the module's reset button, to recover the system prompt.

(19) Dump Disk - DD

The following command was entered by an RS-232 terminal or computer connected to the TEST port:

'DDN'

where 'N' was any hexadecimal number from 1 to FA3, and specified the number of sectors to be transferred from the floppy disk to the device attached to the AUX port of the relevant module. The transfer started from the first sector and continued until the required number of sectors had been moved. There was no demarcation between sectors. The data output at the AUX port was in ASCII-coded hexadecimal an could be printed as hexadecimal on a receiving terminal. This routine was arranged to accept X-ON (CTRL Q) and X-OFF (CTRL S) signals from the receiving terminal, or computer, to regulate the flow of data.

4.2 Load Programmer Application Package

Application software was provided for the Load Program Module and for the Data Acquisition Module. The abbreviated flow charts in this report were designed to highlight major functions only, and much detail was omitted. Complete flow charts and annotated source listings were produced and reference should be made to Appendix 3 for additional information. Application software for the Load Program Module was subdivided into three software modules:

- The interactive start-up sequence.
- The excursion generator, program monitor.
- Miscellaneous utilities for peripheral communication and interrupt handling.

4.2.1 The interactive start-up sequence

On power-up, or following a manual system reset the processor automatically restarted at the beginning of this sequence. The messages used by the system to direct the operator, to inform the operator of faults and to log test status were also in this software module.

These messages were set out in Appendix 7 and reference should be made to the flow chart in Figure 2. In this start-up sequence the system tested the validity of data stored in battery-backed CMOS RAM by checking a short RAM sequence against the same sequence stored in ROM.

The program header on the disk currently mounted in the disk drive was printed out to identify this load program to the operator, and the operator could change disks if required. Once the required load program was mounted and confirmed by the operator, an automatic test of feedback gain was made to ensure it was correctly set to produce the loads specified by the program. For this purpose each program header had a Span Data word which set out the command magnitude to be applied to the DAC, and the number of the shunt 'Calibrate' relay to be operated to produce a simulated equivalent load signal. If the feedback null failed to set the feedback gain was assumed to be in error and requiring adjustment. The system printed 'Span Error'.

The formats fo: program header, flight header and turning-point load values were as set out in Appendix 12 which also contained a general format specification for the loads disk. If the controller span was appropriate for the loads disk the next step in the sequence required the operator to select, or reject data logging, according to circumstances and the current status of the test. When data logging was required the start-up software would interrogate the Data Acquisition Module (DAM) to establish its proper operation. When the DAM was on-line the program header would be sent to it for recording on the output data disk.

A test run could start from the beginning of a loads program or from a point reached in an earlier run. Sufficient data to define the point reached, namely turning-point counter, flight count and program count, was stored in battery backed CMOS RAM. However, if a restart was requested this data would be lost with the clearing of the counters. To avoid unintentional loss of test status the operator was asked to confirm any restart request. Once the conditions for restart had been determined the test status was sent to the DAM, if that module was on line, to be recorded on the output data disk.

At power-up the load program disk drive was set to the start of the disk to read out the program header. If a restart was subsequently requested the drive was then correctly placed to read the header of the first flight. If, however, a continuance from a point reached in an earlier run was required the disk had to be searched, first for the required flight and then for the required turning point in that flight. As each flight header was detected the flight number and number of turning points in the flight were transferred to two single-word buffers, FLTBUF for the flight number and TPBUF for the number of turning points. The contents of FLTBUF and TPBUF were then compared with the contents of the counters in CMOS RAM, FLTCNT and TPCNTT, to confirm the required flight had been located. Having located the start of the required flight, sectors were transferred progressively to the RAM buffer until that containing the required turning point was in the buffer. The buffer pointer was then advanced to that turning point, and the system was ready to proceed to the next, and last, step in the start-up sequence.

The system checked external fault indicators and if these were clear it enabled the hydraulics and directed the operator to turn them on. Once hydraulics were applied the 'DWELL' command was issued to the servoloop, the latch safety system was enabled and a STOP flag was set. This terminated the start-up sequence and the first module of the application program.

4.2.2 The excursion generator, program monitor

The second software module had three major sub routines, BEGIN, EXCUR and TERM; and two minor sub routines SGNMP and TSTMD. Reference should be made to the flow chart of Figure 3.

SGNMP used the current and next turning point values to determine whether the next excursion would be positive or negative going, and what the command value would be at the mid-point of this excursion.

TSTMD used the next command increment and the current command value to determine whether this increment would advance the command beyond the mid-point value. It calculated the difference between current command and mid-point value and compared this difference with the value of the next increment. If the difference was greater than the increment the increment was used, but if the difference was smaller the increment was discarded and twice the difference was added to, or subtracted from, the current command. Once the mid-point command value was passed a MIDPOINT flag was set.

When applying the load program the system cycled continuously through BEGIN, EXCUR and TERM. BEGIN before starting the excursion, EXCUR while generating the excursion and TERM when the command has reached the next turning point value.

BEGIN when entered either from the Start-Up Sequence or from TERM tested the STOP flag. This flag was set in the start-up sequence and might be set between one excursion and the next. See Comments made under TERM. When the STOP flag was found set message MI8 was printed requiring an operator response in under ten seconds to avoid automatic shutdown. The options available to the operator were:

(1) Continuous cycling

(2) A single excursion with a hold at the next turning point.

(3) A waiting state with no action by the system.

(4) Automatic shutdown on timeout.

Details of message MI8 and all other messages were set out in Appendix 7.

If an appropriate operator command was received, or the STOP flag was not set, the next turning point value was read from the load command buffer and the buffer pointer advanced to the following load. If this pointer now indicated the first address beyond the end of the buffer a SECTOR flag was set.

Following the end of buffer test, a test for control loop null was made and continued until the null was found to be set or the time provided for system null expired. If the null did not set within the prescribed time SHUT DOWN was automatic. If the DAM was in use data readings were made, whether the null was achieved in time or not, before any other action occurred. Each time a set of data readings were made it was prefixed with the load command value being attempted.

With the null achieved the next turning point value was used to calculate excursion direction and mid-point command value. The DWELL command was removed and the system entered the excursion generator EXCUR.

EXCUR generated the series of command steps which together made up an excursion and which, when output through a DAC, became the input voltage command to the analogue load control servo loop. The excursion was created by adding command increment values to, or subtracting them from an accumulated value in an input buffer feeding the DAC.

Two increment types were used in generating the excursion, those which were stored in a table in ROM and used to produce the non-linear beginning and ending of all excursions, and those increments which had a single constant value and were used to generate the ramp section of the excursion. The system generated the excursion by first taking the table increments one at a time, and then the fixed value increment over and over again until the pre-calculated mid-point value of the excursion was reached. Each time any increment was transferred to the DAC buffer the event was counted until the mid-point had been reached. Beyond the mid-point the increments were transferred as before but the counter was decremented for each transfer until it reached zero. This count provided the information telling the system when to switch from fixed value elements back to table elements to terminate the excursion.

The rate at which increments were transferred to the DAC input buffer was controlled by a hardware timer which could be set to achieve a desired excursion slew rate. The timing could be arranged if necessary to yield a slower slew rate when loading changed sign, from tension to compression for example. The calculations to determine the hardware timer settings for a particular slew rate were discussed in Appendix 6.

Once the increment counter was back to zero the system left EXCUR and entered TERM.

TERM set the DWELL signal and then performed the operations discussed below. Reference should be made to the flow chart of Figure 4.

(1) Advanced the turning point counter by one and compared the count with the number in TPBUF to test for end of current flight. If the end-of-flight had been reached the flight counter was advanced and a similar test made for end of program. When the flight counter was advanced the turning point counter was cleared, and when the program counter was advanced the flight counter was cleared. When the end-of-flight condition was detected a check of the TERMOR flag had to be made together with a printout of the current status of the test. If the TERMOR flag was set the current flight number had to be compared with a flight number stored in TRMFLT buffer to see if the test was to stop at this time. If this was the terminating flight the STOP flag was set.

(2) Tested the SHUTDOWN flag to see if the last excursion was to zero load. If this flag was set the system turned off the hydraulics and, if required, read a further set of output data on the DAM. If the SHUTDOWN flag was not set a test for input from the keyboard was made. Keys could be struck at any time, but the signals were decoded only at the completion of an excursion, and then only two commands would be accepted by the system. These commands were:

- HD (Hold) - would stop the system at the current turning point.
- TXX (Terminate) - would set the TERMOR flag and store the flight number XX in TRMFLT buffer.

Both these inputs had to be terminated by Carriage Return. The check for 'HD' was made at every turning point but the TERMOR flag was checked only at end-of-flight. If 'HD' was current, or the current flight was that stored in TRMFLT, the STOP flag was set.

(3) Checked the SECTOR flag, and when it was set cleared it before reading another sector from the loads disk into the loads buffer. The buffer pointer was reset to the start of the buffer. At the end of each flight, which might occur in the middle of a sector, another sector was transferred to buffer and the buffer pointer reset. The SECTOR flag was ignored in this case.

(4) Handled the end of a program as for the end of a flight except that here both turning point and flight counters were cleared.

(5) Minimised the possibility of disk reading errors by reading each sector twice into separate buffer and comparing the results. If there was disagreement the process was repeated. Refer 4.2.3.(10).

After completion of all actions found necessary in TERM the system returned to the start of BEGIN and the process was repeated.

4.2.3 Miscellaneous utilities

The third software module contained eleven support routines as follows:

(1) A routine to print directions and fault messages on a teletype. When switched to a second UART this routine directed commands and data to the DAM.

(2) Routines to handle interrupts. Interrupts were generated by the two UARTS in both sending and receiving modes, and by the hardware timer on completion of preset periods. The keyboard and DAM signal interrupts employed the same handler. A separate interrupt handler was provided for the timer.

(3) A UART control routine which, by employing the printer and keyboard routines and the interrupt handler, controlled the flow of messages to and from the teletype and DAM.

(4) Keyboard routines were used to control acceptance of characters from the teletype keyboard, to decode these characters and to provide suitable flag signals to the other software modules.

(5) Hardware timer routines permitted the system to select four different delay periods and to signal other routines when the selected period had expired. Timing was important in excursion generation, waiting for the control loop to null and waiting for operator responses to system requests.

(6) A routine used to reset the DAC and release shunt selecting relays after automatic system 'Calibrations'.

(7) A routine to convert the flight and program headers recorded in ASCII on the load disk into binary data for system use.

(8) A test-status-log routine which converted binary data to ASCII. This routine provided current-turning-point, flight and program numbers, time-of-day and date in ASCII for printing on a teletype and recording on DAM disk. It also provided an ASCII version of the failing command i.e. the command presented to the DAC at the time of latch interrupt. This failing command was retained in Register R5.

(9) A routine to check the flight header and to transfer data in the header i.e. flight number and number of turning points in flight to flight buffer FLTBUF and turning-point buffer TPBUF.

(10) A routine to read load turning-point data from the loads disk, one sector at a time, into RAM buffers. This routine called TDOS facility 'RD' to perform the reading. To minimise the possibility of an erroneous transfer of turning point data a double reading was made into two buffers, and the contents were compared word by word. If a disagreement was detected the double reading was repeated. By incrementing a special counter each time a double reading had to be repeated, a measure of the quality of the disk and disk drive could be obtained. After three double readings of a sector without agreement between the two buffers the system printed 'Disk Error' and initiated system shutdown. This routine also tested the disk status register to ensure the disk reading had been completed without error in the data read. If the status register was not zero the 'Disk Error' message was printed and the system shutdown as before.

(11) The safety latch interrupt handler was entered when the safety latch tripped. The latch could be tripped by the operator activating the Emergency stop buttons, or by signals indicating one or more of the following parameters had exceeded their set limits:

Servo Error

Applied Load

Actuator Travel

To minimise spurious interrupts arising from electrical transients this software checked the duration of the interrupting signal. If it was less than six milliseconds the interrupt was regarded as spurious and cancelled. If it was longer than six milliseconds the system shutdown rapidly with hydraulics off and actuator locked. A failing-command log was printed and, when required, data was record by the DAM.

4.2.4 Operation with an environmental controller

To make provision for the use of an environmental controller with the load controller a flag was inserted in the flight header extending this header to five words. Refer to Appendix 2. If this header flag was set the program would:

(1) Set an environmental flag ENVFLG to indicate an environmental change was required at the end of this flight.

(2) Set the TERMOR flag and load TRMFLT buffer with the current flight number. Any current terminating conditions were saved elsewhere.

At the end of the flight, when the program detected a terminating condition with the environmental flag set, a signal was sent to the environmental controller requiring that controller to change the system to the next pre-set environmental state. The load controller waited for whatever time was required to achieve the new environment and then printed the end-of-flight log. It then proceeded with the next loading flight in the sequence.

If the operator requested a stop at the end of a flight for which an environmental change was required that stop would be ignored.

5. THE DUAL SYSTEM

It was possible to couple together a number of stand-alone controllers in a single master, multiple salve configuration. However this arrangement was regarded as a means for the operation of only a small number of single channel stand-alone controllers in parallel, and not as an appropriate solution to the more general multi-channel system requirements. For even a small number of controllers the arrangement to be described was regarded as an interim rather than a final solution.

The two-channel (dual) systems were necessary wherever bi-axial loading was required, and to date two such systems have been produced, one for aircraft component testing and one for fibre composite tests.

5.1 The Dual Configuration

The dual controller employed two complete single channel stand-alone controllers in a single cabinet. One controller was designated Master and the other functioned as a Slave. Both controllers were connected to a common bus by, respectively, Master and Slave mode modules. Conceptually any number of Slave controllers could be connected through individual Slave mode modules to this bus. However, in practice operational problems would be likely to impose severe limits on the number of Slave units used.

The Data Acquisition Module (or modules if more than eight data sensors were required) was connected to the Master Load Program Module by an RS-232 cable between the AUX port of the Master and the TTY port of the DAM. Where several DAM's were required they were connected in 'Daisy Chain' fashion, AUX port of preceding module to TTY port of following module. The simple toggle switch located in each module should only be turned on in the last module to indicate it was the end of the chain.

The operator controlled the dual system by a single terminal connected to the TTY port of the Master module. Because both controllers were constrained to keep in step, as discussed later, progress logs for the Master unit were indicative of the progress of the complete system. However, only faults arising in the Master unit could be reported directly by the teletype. If faults occurred in the Slave unit the system would stop and go to a safe reset state, but to identify the fault causing reset it was necessary to examine error flags in the Slave unit's RAM memory.

5.2 Synchronisation

To maintain synchronism between two cycling controllers, the load turning-point programs for both had to contain the same number of turning points although the magnitudes of corresponding turning points could be quite different. To meet this requirement it could be necessary for one or the other controller to execute a sequence of turning points all at constant amplitude. The software was arranged to meet this requirement.

5.2.1 Initialisation

During initialisation both Master and Slave units moved through their respective start-up sequences at their own rate, reporting errors, whenever they were encountered, until the automatic shunt 'Calibration' procedure was reached. Each controller could select one of two shunting resistors but the selecting relays were controlled from the Master unit. the Slave unit waited for the Master to operate this relay, after which it applied the command given in the header of its own load disk. If the wrong disk had been mounted in Master or Slave the test for null at this point would probably fail, proper null would not be achieved, and the calibration command would not be cleared. Failure to clear this command would prevent the Slave unit proceeding further through its start-up sequence. If the automatic calibration checks were completed satisfactorily the two units then searched their load disk for the required starting point loads. If the Slave unit failed to find its required starting point within a limited time after the Master had located its starting load, the latter would report 'Hydraulics or Slaves Disabled' and shut down. Otherwise, if both units were ready, the Master unit would enable hydraulics and wait for the operator to apply hydraulic power.

5.2.2 Load cycling

Synchronisation of the load excursions of Master and Slave units was achieved by four signals:

NULL

ACK

MASTER DWELL

SLAVE DWELL.

NULL was a hardwired signal and tested by the Master unit only. This NULL signal was asserted only when both Master and Slave control loops had brought respective feedback and monitor signals to the null condition.

The remaining signals were software controlled.

ACK was asserted by the Slave unit and tested by the Master. ACK was set whenever the Slave unit was ready to execute another excursion. It was cleared as soon as the excursion was completed and SLAVE DWELL set. Once ACK had been cleared the Slave unit proceeded to check status, read a sector from disk where necessary, and fetch the next turning point value. It would then set ACK again.

MASTER DWELL was set by the Master unit after it had completed an excursion and the Slave unit had cleared ACK. Refer to the flow chart of Figure 5 for a simplified representation of the operating sequence for the Master and Slave units. Before starting an excursion the Master unit had to see with MASTER DWELL and SLAVE DWELL set (i.e. DWELL ERROR not set) the NULL signal set and the ACK set. Once these conditions were established the Master unit could remove MASTER DWELL and excursion to the next turning point. Here it waited for the Slave unit to remove ACK before applying MASTER DWELL and preparing for the next excursion.

Before starting an excursion the Slave unit waited for MASTER DWELL to be turned off indicating the Master unit was about to start an excursion. The Slave unit could not set ACK to signify its readiness to perform another excursion until MASTER DWELL was set.

The MASTER DWELL, SLAVE DWELL and ACK signals were so related that the units had to excursion together.

6. THE DATA ACQUISITION SYSTEM

A major design objective of this system was to avoid some of the problems associated with the use of the shunt 'Calibration' technique as the basis for the determination of the physical quantity measured (Reference 4). To this end the shunt 'Calibration' was used only as a check on total channel repeatability. The conversion factor used to obtain physical quantity from the measured signal was arranged to be a function of sensor excitation, sensor 'gauge factor' and amplifier gain only. Care was taken in the selection of components and in circuit design to ensure these parameters remained stable.

As previously indicated one, or several of these systems could be used alone or with a stand-alone controller. Each data acquisition system comprised:

- Data Acquisition Module
- Eight Sensor Amplifiers
- Amplifier Termination Box

6.1 Data Acquisition System Hardware

6.1.1 Data acquisition module

Physically this module was a copy of the Load Program Module with an internal power supply added, the CMOS memory card removed, and the rig interface card replaced by a data acquisition card. The added complexity of an internal power supply was justified because:

- It permitted independent operation of one or more of these modules.
- It ensured adequate power would be available irrespective of the number of modules used.

6.1.2 Data acquisition card

This card was designed to accept eight ±5 volt analogue inputs; to digitise these inputs and to transfer them to the system bus. Differential analogue signal inputs were provided, but common mode levels were limited to within six volts of analogue common to satisfy the specification of the hybrid data acquisition system (SMD 857 KG). The eight inputs were multiplexed to the sample-and-hold amplifier driving the analogue-to-digital converter (ADC). This ADC had 12 bit resolution (about 0.03 per cent full scale), a conversion rate of about 20 000 single channel readings per second, and an output code format of 2's complement sign extended, positive true binary. The digital output of the ADC was tri-state buffered to the bus. Information for the input multiplexer was written to the card at the same bus address as that used to read the ADC output, namely 8016_{16}. Writing to this address initiated a conversion which, when completed, caused the most significant bit of the status register to be set low. This status register was accessed by reading memory location 8014_{16}.

Commands were issued by writing to 8014_{16}. In particular shunt 'Calibrations' were asserted by setting the most significant bit high at that location. This caused the calibrate control line to be driven to a value five volts below the amplifier supply line thereby energising the calibrate relay.

The remaining circuit on this card contained regulators providing ±8 volts for the sensor amplifiers.

6.1.3 Sensor amplifiers

Each amplifier was mounted on a small printed circuit card designed to be attached to the test structure as close as possible to the active sensor. Each card contained an instrumentation amplifier, bridge completion resistors, a shunt 'Calibration' circuit and a sensor supply.

For a given sensor configuration the ratio of physical quantity to amplifier output voltage could be adjusted in steps by means of a link, and continuously by varying the sensor supply voltage. This supply could be varied over a wide range by adjusting Rvb, a supply resistor on the amplifier card. (Drawing No. 59557-A2). By providing a low supply voltage to these amplifier cards power dissipation was kept to a minimum, and stability and reliability enhanced.

By adjusting sensor volts and amplifier gain it was possible to make the amplifier output direct reading in the physical quantity being measured. For example with strain sensors a resolution of one microstrain and a range of +2047 to -2048 microstrain could be achieved. The shunt 'Calibration' was used as a check on system repeatability only. The resistor used was selected to give a positive output signal at the output of the amplifier of about half full scale (i.e. 2.4988 volts). The amplifier card also provided space for fixed initial-balance-adjusting resistors, whereby partial initial balance of the sensor bridge could be achieved.

6.1.4 Amplifier termination box

Each of the eight amplifiers of an installation was connected, by a six-wire shielded cable to a screw-terminal strip in this termination and power distribution box. The box was connected to the data Acquisition Module by a 25-core flat ribbon cable. This box facilitated the installation, adjustment and trouble-shooting of sensor and sensor-amplifier circuits.

6.2 Data Acquisition Software

Although this software package was relatively straightforward it provided for the operation of Data Acquisition Modules either singlely or in 'daisy-chained' groups, with control from a Load Program Module, terminal or master computer.

The system facilities package was exactly the same as that used in the Load Program Module. Disregarding the general housekeeping functions of the application package the remaining software accepted and executed four incoming commands and read, tested and stored sensor data. Source listings and flow charts have been prepared and the sixteen software routines provided were listed in Appendix 8.

Both input commands and recorded data were required to conform to the format specification given in Appendix 2. This format was chosen to ensure data recorded on disk could be recovered for presentation and further analysis with minimum difficulty.

6.2.1 Input commands

These commands were in ASCII and reached the Data Acquisition Module through its TTY port. Each command or input data message had to be terminated by '#'. In the present system all incoming commands and data arose in the Load Program Module. The commands and responses were as follows:

I# This was the interrogate command. It caused the DAM to read and save as zero values the current sensor outputs, and to perform a shunt 'Calibration' of all sensor channels. It also brought about an automatic determination of the number of active sensor channels (which must be less than or equal to eight), and tested all active channels for sensitivity. To facilitate the determination of the number of active channels all active sensors had to be located on consecutive positions, starting from position one, on the termination box. All unused inputs required 'Reverse' cal. circuits to be connected.

R:XXXX,# In this command XXXX was the applied load command in ASCII. This command required the DAM to read all active sensor channels, correct the readings for initial zero values and store the results. These results were recorded in units of the physical quantity being measured. Each sensor data record on the disk had a header containing the number of active sensors being recorded and a hexadecimal representation of the DAC load command at which the measurement was made. This input command could also have the form R# with no load command value. Dummy load command values were inserted automatically for this form of command.

<TEST MESSAGE # When the ASCII '<' was received the DAM would regard all the following characters prior to '#' as ASCII, and store them as a message on disk. This command was used to transfer test logs and time-of-day clock outputs onto disk.

T# This command was used to inform the DAM that no further readings would be required, and that any data currently held in buffer memory should be transferred to the disk.

6.2.2 Data storage

Data received as input messages, or as corrected sensor readings, was accumulated in two 256 byte RAM buffers, and as each buffer was filled its contents was automatically transferred to the floppy disk. To conform with the format rules of Appendix 2 each sector had to start with a number from 1 to 8 or the ASCII '<'.

Where an incoming message overflowed from one buffer to the other the message was terminated and a new header inserted at the sector boundary. In the case of the sensor data the length of each record was known and if the next reading would overflow the buffer boundary the current buffer was closed and the record written into the empty buffer. Each sensor data record had the three-byte header containing number of active channels and the load command value.

The available capacity on the floppy disk was limited, and continuous record of a full eight channels of sensor was possible only for very short runs. Software was provided to ensure that when recording was restarted after an inactive period the first DAM record would be located at the start of the sector immediately following the sector used to terminate the previous run. It was important to reset the DAM whenever another disk was used to ensure a search for the end of any earlier record was made, and the sector counter was properly initialised.

6.3 Support Hardware

The two units to be briefly described in the following paragraphs were not considered to form part of the Data Acquisition System. They were of the nature of tools, and used in the trouble shooting, setting up and standardisation of individual systems.

6.3.1 Bridge amplifier calibration and test box

This test box contained a wheatstone network for connection to the input of the sensor amplifier. Other circuits within the box could be connected directly to the output of a sensor amplifier or indirectly to this output at the 25 pin outlet of the terminal box. The box simulated the functions of the Data Acquisition Module, and its purpose was to facilitate the adjustment of amplifier gain and bridge voltage to achieve direct reading. It was also an aid in the selection of initial balance and shunt 'Calibration' resistors, and in the diagnosis of sensor circuit faults.

6.3.2 Data acquisition module test box

This test box connected directly to the Data Acquisition Module in place of the Amplifier Termination Box. Its purpose was to permit checkout and maintenance of the Data Acquisition Module without the need for sensor circuitry. Signals equivalent to sensor amplifier outputs for both normal and shunt 'Calibration' conditions were provided on eight signal lines. Both normal outupt and shunt 'Calibration' signals were adjustable over a considerable range on all channels; and test points for the connection of a digital voltmeter were provided for each channel to facilitate the adjustment of these signals to desired values.

The shunt 'Calibration' signal could be initiated from the Data Acquisition Module or by means of a manual press-button on the front of this test box.

7. CONCLUSION

Experience with several of these controllers showed the level of consistency in writing to and reading from the floppy disks was lower than that considered desirable. This inconsistent performance was thought to be associated with the operation of the disk controller, but efforts to date have failed to pin point the source of the trouble.

With a better understanding of the length of load spectrum likely to be required, and a recent increase in the storage capacity of EPROM chips the future replacement of floppy disk storage by solid state store appears very likely. This form of storage, with no moving parts, should be faster and more trouble free than the existing disk units. The design objectives set out in Section 2 were substantially met. However, while these controllers permitted a limited study of possible multiple systems it was clear they were too complex and expensive for use in the 'many-channel' control applications.

Developments in the electronics industry in general, and in the microcomputer industry in particular, rendered the designs used for this system obsolete almost before they could be brought into production. With the components currently available much could be done to improve the design for single and dual controllers. However the experience gained with these controllers provided a clearer insight into the technical criteria which must be met for future, all-digital, multi-channel systems.

8. REFERENCES

1. 9900 Family Systems Design and Data Book. Texas Instruments Inc., 1978.

2. C.J. Ludowyk. Operators Manual for ARL Servo amplifier. ARL-STR-TM-407, 1985.

3. E.S.Moody and I.Powlesland. The Load Sequ·nce Controller Family, an Operators manual. ARL-STR-TM (in publication), 1986.

4. E.S.Moody. The Constant Current Strain Gauge Bridge. ARL-STR-TM-385, 1984.

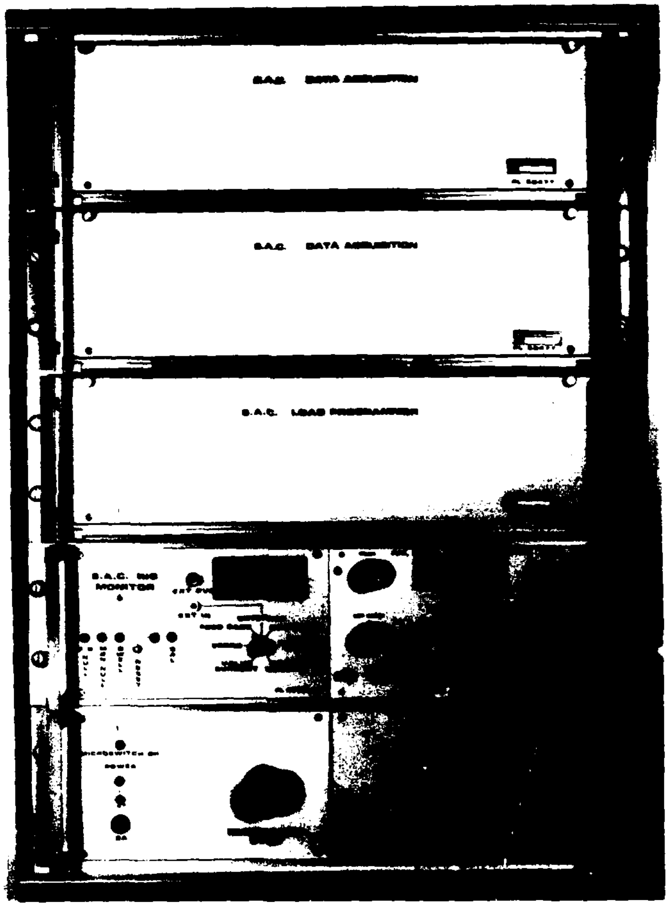
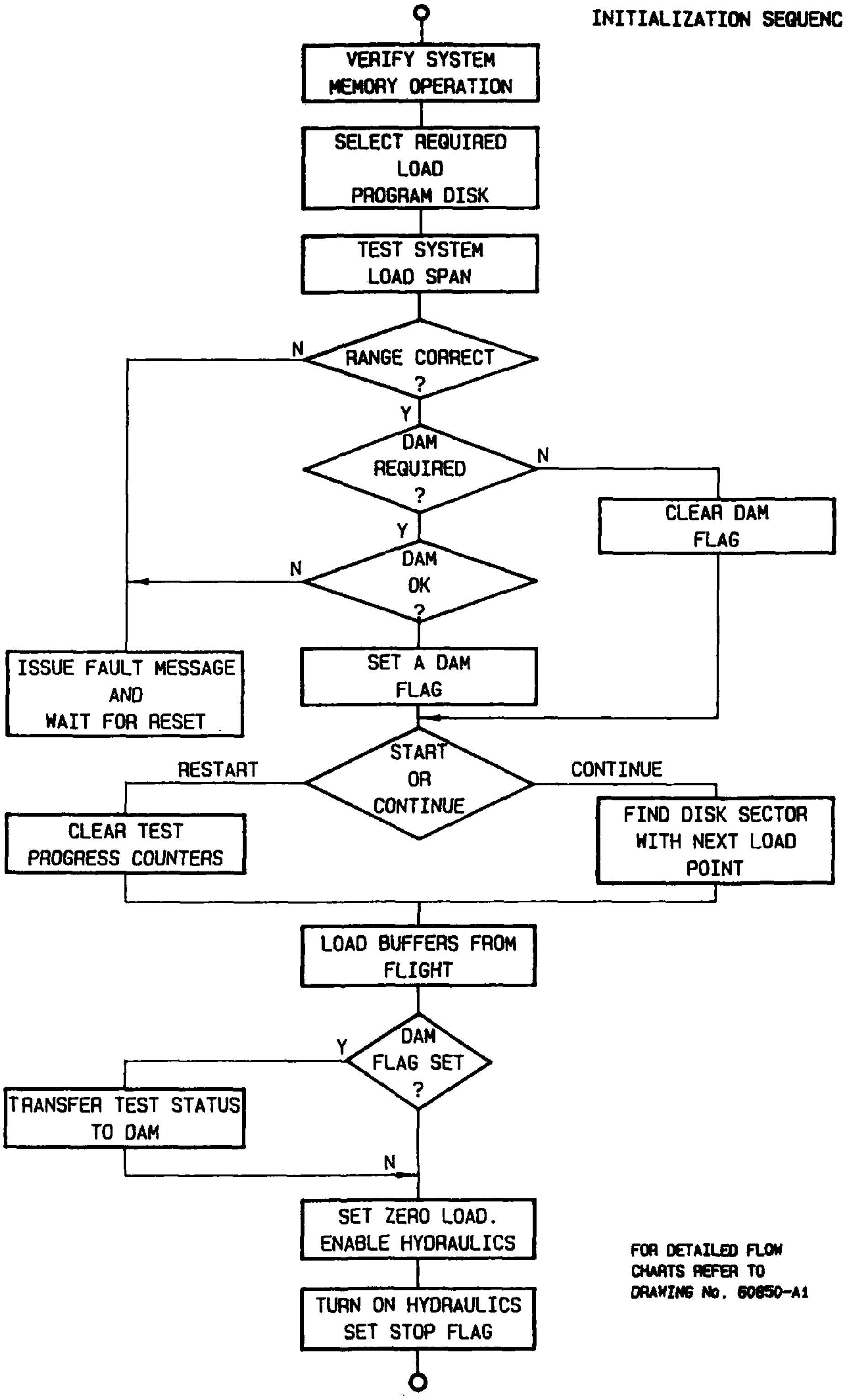


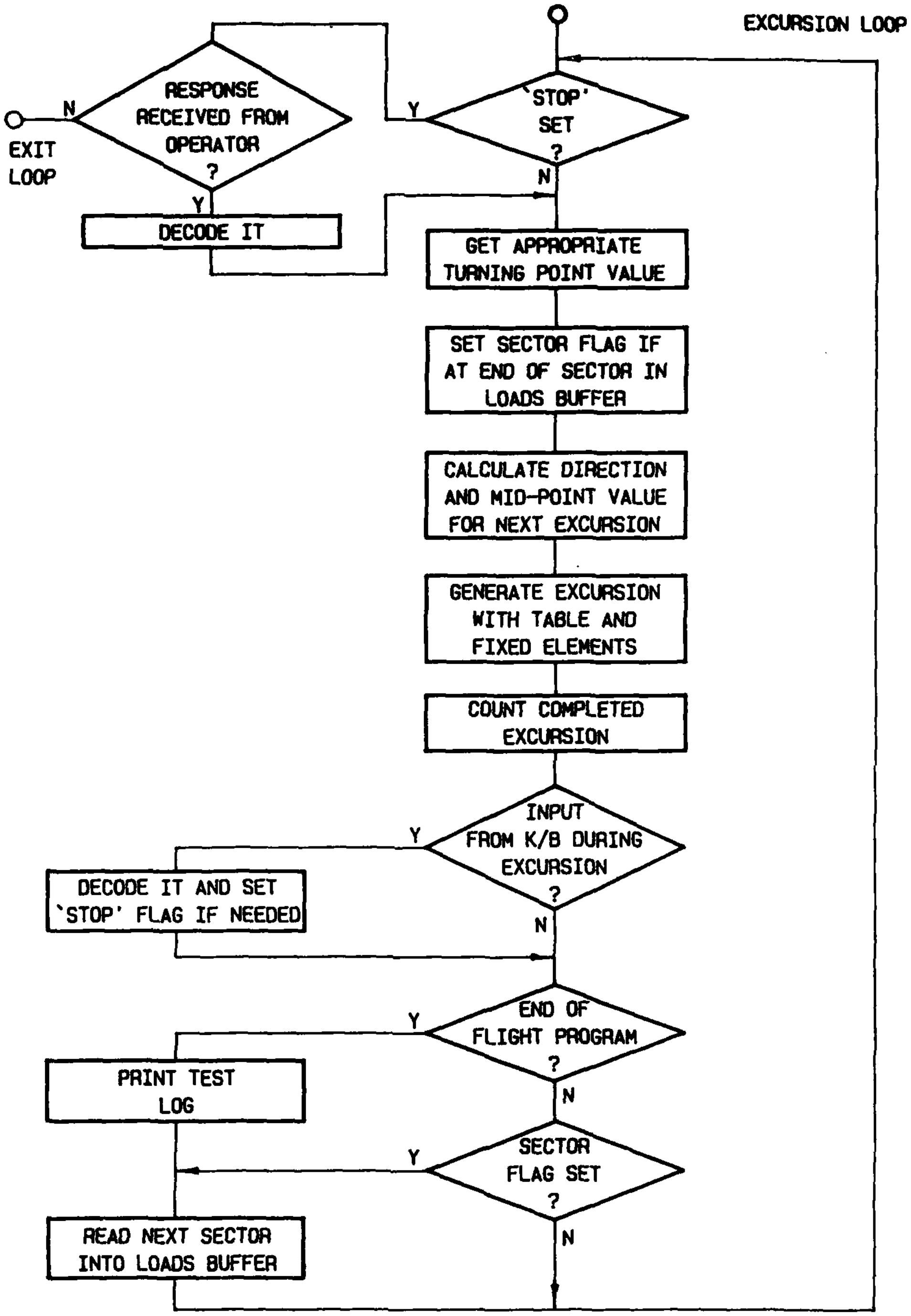
FIGURE 1. SINGLE CHANNEL STAND ALONE CONTROLLER WITH 2 DATA ACQUISITION MODULES

INITIALIZATION SEQUENCE


VERIFY SYSTEM MEMORY OPERATION
SELECT REQUIRED LOAD PROGRAM DISK
TEST SYSTEM LOAD SPAN
RANGE CORRECT ?
N
Y
DAM REQUIRED ?
N
CLEAR DAM FLAG
Y
DAM OK ?
N
ISSUE FAULT MESSAGE AND WAIT FOR RESET
SET A DAM FLAG
START OR CONTINUE
RESTART
CONTINUE
CLEAR TEST PROGRESS COUNTERS
FIND DISK SECTOR WITH NEXT LOAD POINT
LOAD BUFFERS FROM FLIGHT
DAM FLAG SET ?
Y
TRANSFER TEST STATUS TO DAM
N
SET ZERO LOAD. ENABLE HYDRAULICS
TURN ON HYDRAULICS SET STOP FLAG

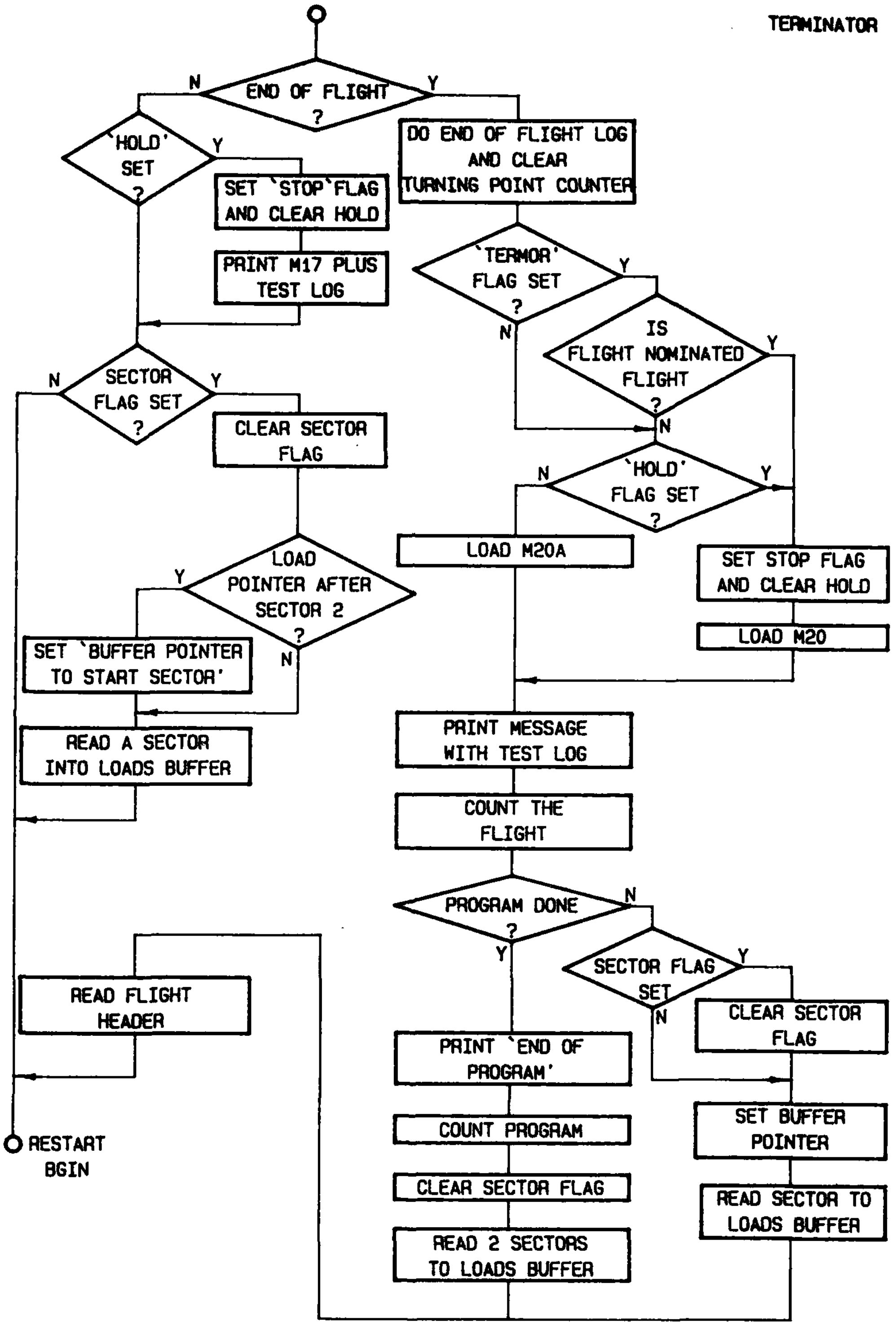
FOR DETAILED FLOW CHARTS REFER TO DRAWING No. 60850-A1

FIGURE 2.


EXCURSION LOOP
RESPONSE RECEIVED FROM OPERATOR ?
N
EXIT LOOP
Y
DECODE IT
Y
'STOP' SET ?
N
GET APPROPRIATE TURNING POINT VALUE
SET SECTOR FLAG IF AT END OF SECTOR IN LOADS BUFFER
CALCULATE DIRECTION AND MID-POINT VALUE FOR NEXT EXCURSION
GENERATE EXCURSION WITH TABLE AND FIXED ELEMENTS
COUNT COMPLETED EXCURSION
INPUT FROM K/B DURING EXCURSION ?
Y
DECODE IT AND SET 'STOP' FLAG IF NEEDED
N
END OF FLIGHT PROGRAM ?
Y
PRINT TEST LOG
N
SECTOR FLAG SET ?
Y
READ NEXT SECTOR INTO LOADS BUFFER
N

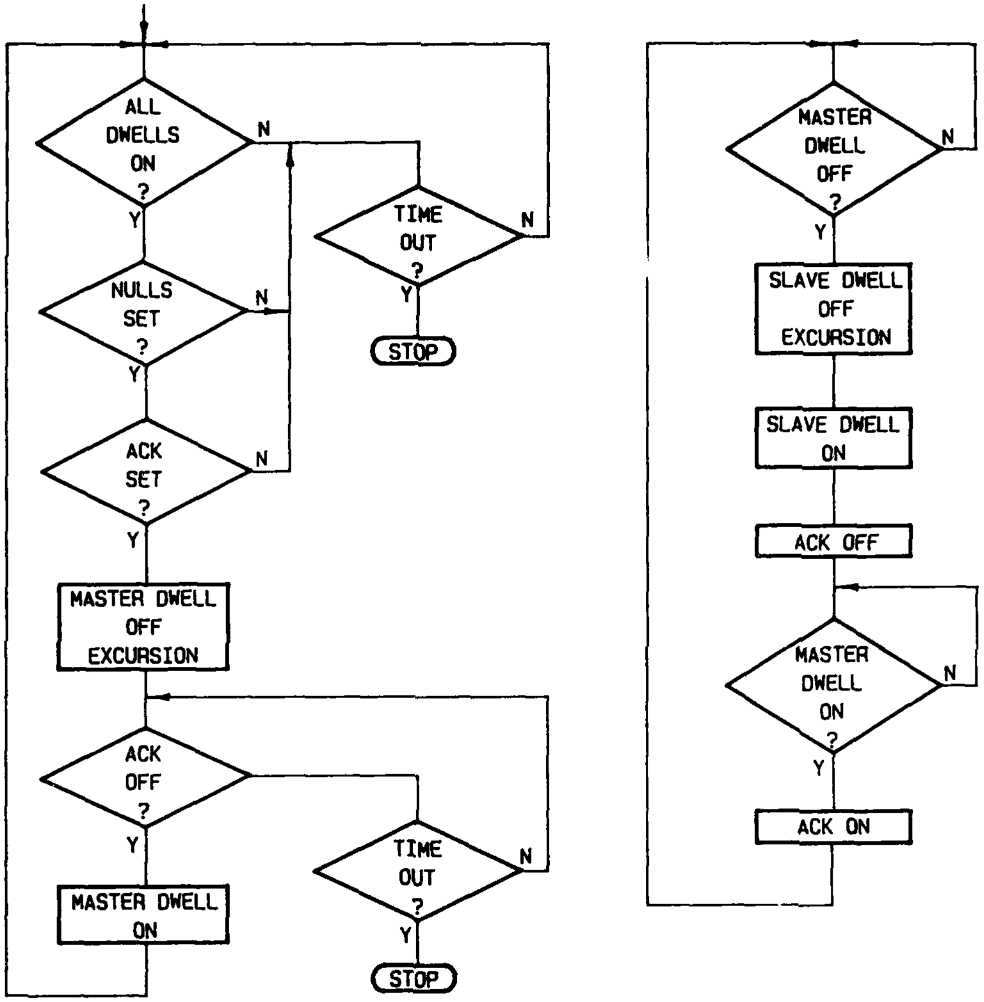
FOR DETAILED FLOW
CHART REFER TO
DRAWING No. 60652-A1
AND 60653-A1

FIGURE 3.



(ALSO REFER TO DRAWING No. 60854-A1)

FIGURE 4.



MASTER CONTROLLER

SLAVE CONTROLLER

FIGURE 5.

APPENDIX 1

PROM PROGRAMMES

IC16 (MSB), 18 (LSB)	2716 TDOS (same for all first generation units)
IC15 (MSB), 17 (LSB)	2716 Relevant application program
IC14	DM7578/N82S123/74S288 The D.A. Unit has no CMOS ram at 1000_{16} and needs some of the cpu card ram at this location for its workspaces. Hence:

	Load Programmer Controller Master Slave	Data Acquisition
Address	Data	Data
00	F9	F9
01	F3	FF
02	FF	F3
03	FF	FF
04	FF	FF
05	FF	FF
06	FF	FF
07	FF	FF
08	FF	FF
09	FF	FF
0A	FF	FF
0B	FF	FF
0C	FF	FF
0D	FF	FF
0E	FF	FF
0F	FF	FF
10	FF	FF
11	FF	FF
12	FF	FF
13	FF	FF
14	FF	FF
15	FF	FF
16	FF	FF
17	FF	FF
18	FF	FF
19	FF	FF
1A	FF	FF
1B	FF	FF
1C	7B	7B
1D	7B	7B
1E	FA	FA
1F	FA	FA

APPENDIX 2

DISK FORMATS

Both load and data disks were 8 inch and used IBM System 34, 256 bytes/sector, double sided, double density format. This could be written on the disk using the 'FD' command.

The format of each track being:

Number of bytes	Hex value of byte written	
80	4E	
12	00	
3	F6	
1	(Index Mark) FC	
50	4E	
		WRITTEN 26 TIMES
12	00	
3	F5	
1	(ID Address Mark) FE	
1	(Track Number) 0–4C	
1	(Side Number) 0–1	
1	(Sector Number) 1–1A	
1	(Sector Length) 01	
1	(2 CRCs) F7	
22	4E	
12	00	
3	F5	
1	(Data Address Mark) FB	
256	DATA	
1	(2 CRCs) F7	
54	4E	
598 (approx.)	4E	Until index mark encountered.

NB.

1. Unformatted the disk capacity was about 1.6 M. byte, formatted, slightly under 1 M. byte.

2. Track 0 is outside track.

3. The 'block' number used by the facilities package was allocated as follows:

Block	Sector	Side	Track
000	01	0	00
001	01	1	00
002	02	0	00
003	02	1	00
004	03	0	00
–	–	–	–
$FA3_{16}$	$1A_{16}$	1	$4C_{16}$

ie. Block 4004 - Sector 26, Side 1, Track 76.

LOAD PROGRAM DISK

Each disk contained only one program, which was described on a label attached to the top surface of the protective jacket. The disk must never be removed from this jacket.

The first sector of the first track on side 0 was reserved for the Program Header. This was written in ASCII according to the following format, with each line terminated with a Carriage Return.

```
<LOAD CONTROL PROGRAM NUMBER xxxxxx
MAXIMUM LOAD EXCURSION xxxx NEWTONS (or pounds)
PRGM TYPE xx
SPAN DATA xxxxx
FLT COUNT xxx
T/P COUNT xxxxx
#,LF,ESC
```

<,#,LF,ESC	Were necessary only if data logging was used.
Load Control Prog. No.	Was an identification number for the disk, eg. 6 characters representing the date.
Max. Load	Amplitude of largest load on disk.
Prog. Type	Type of spectrum, eg. Block, Static-Cal., Fatigue, etc.
Span Data	Decimal representation of 16 bit word, with lower 4 bits selecting the cal. relay, and upper 12 being fed to the DAC.
Flt. Count	Decimal number of flights in this program.
T/P Count	Approximate average number of turning points in a flight.

Each 'program' was divided into blocks of turning points referred to as 'Flights'. Each flight commenced with a 10 byte header, which was always at the start of a sector, the first one being at the start of sector 2. As each turning point required 2 bytes, up to 123 turning points could be stored after the header (in the first section), and up to 128 in each subsequent sector as required. The last turning point should have a zero load value, and not be the first or last word in the disk sector.

The Flight header is written in hexadecimal and contains the following 5 words (10 bytes):

FFFF	Header label
FFFF	Header label
xxxx	Sequential number of the flight (starting from 0)
xxxx	Accurate number of turning points in this flight
0000/FFFF	Environmental flag, used only if environment control was linked to flight count.

SENSOR DATA DISK

Data was placed on this disk according to the following rules:

1. Only two types of fields were valid:

   Text Message..with a '<' header and '#' terminator.

   Sensor Data...with a '1'xx-'8'xx header, which defined the length of the field (2n + 3 bytes), and current load command (xx or, if not supplied FFFF by default).

2. A field has to be entirely contained in one sector. For convenience a text string could be broken into 2 individual text messages at the sector boundary, by finishing one section with a '#' and beginning the next with a '<'.

3. Text strings should not exceed 256 characters in length.

4. Fields were to be placed on the disk starting from the first position in the first sector, filling that block until there was insufficient room for the next complete field, then positioning that field at the start of the next sector.

5. Prior to being used the disk had to be formatted with data that did not constitute a valid header (ie. not '1'-'8' or '<'). this was to allow the end of the record to be located (ie. first sector not beginning with a valid header).

APPENDIX 3

RELEVANT DRAWINGS

Hardware

58 457 A1	S.A.C. System Block Diagram
58 455 A2	S.A.C. System General Assembly
PL 58 454	S.A.C. System Parts List
58 456 A0	S.A.C. System Family Tree

58 386 A1	S.A.C. Rig Monitor Card Circuit Diagram
PL 55 474	Mk.2A Servo Amplifier Parts List
55 456 A1	Servo Amplifier Mk.2A Block Diagram
55 455 A0	Servo Amplifier Mk.2A Interconnections
58 230 A1	S.A.C. Hydraulics Controller circuit
PL 58 338	S.A.C. Load Programmer Parts List
58 340 A1	S.A.C. Load Programmer Circuit
PL 58 477	S.A.C. Data Acquisition Parts List
58 479 A1	S.A.C. Data Acquisition Circuit
PL 58 488	S.A.C. Data Acquisition Power Supply P.L.
58 493 A1	S.A.C. Data Acquisition Power Supply Cct

59 560 A2	System Test Module
PL 59 490	Termination Box Parts List
59 491 A2	Termination Box Circuit
PL 59 556	Mk.3 Bridge Amplifier Parts List
59 557 A2	Mk.3 Bridge Amplifier Circuit/Assembly
59 565 A2	mk.3 Bridge Amplifier Calibration Module

59 748 A1	Intersystem Bux, Interconnection & Cct
60 096 A1	System Accessories (Mounting Hardware)

Flow charts, Control

60 850 A2	System Start Up Sequence
60 851 A3	Disk Associated Routines
60 852 A2	Excursion Starting Routine
60 853 A2	Excursion Generator
60 854 A2	Excursion Terminating Routine
60 855 A3	Flight and Excursion Parameters
60 856 A3	Test Status Log
60 857 A3	Test Status Routines
60 858 A2	Test Clock Handler
60 859 A3	Test Status Routines
60 860 A2	Interrupt Handler for DAM, Printer & Keyboard
60 861 A3	Keyboard Routines
60 862 A3	Keyboard Data Decode

Flow charts, DAM

60 863 A3	Console Routine
60 864 A3	Seek Start
60 865 A3	Interrogate Routine
60 866 A3	Read Strain Data
60 867 A3	Reading Data Acquisition Module
60 868 A3	Terminate Current Run
60 869 A3	Taking Data Message
60 870 A3	Transfer Data to DAM Disk
60 871 A3	Echo and Finish Up

APPENDIX 4

SYSTEM PARAMETERS

The TEST port was automatically active when a terminal was plugged into it, and the RESET button pushed. The baud rate of this terminal had to be 4800 or less and was calculated by the system on receiving a upper case 'X' not less than 2 seconds after resetting. Typing 'HE' gave a list of available commands. Most system parameters could be checked, and if necessary altered, using the 'AM' command.

In the Load Program Module certain constants needed checking prior to operation, these were:

1008_{16} Timer for Cal and response time (normally $7FFF_{16}$)
$100A_{16}$ Zero crossing rate word (see Table, typically $007B_{16}$)
$100C_{16}$ Non zero crossing rate word ($007B_{16}$)
1003_{16} Another times (try 0003)

Load disks should be checked for spurious first characters.

The error message: 'RAM SUPPLY FAILED. GET HELP' indicated the battery backed CMOS ram had lost its reference values. These were 1000_{16}-1111_{16}, 1002_{16}-2222_{16}, 1004_{16}-4444_{16} and 1006_{16}-8888_{16}.

The error message: 'DISK READING ERROR. GET HELP' indicated the teletype should be connected to the 'TEST' port. Address 1040_{16} inspected to see which record was the subject of the read attempt, 1042_{16} checked for destination or source area in memory, and 1044_{16} checked for the type of error. The types of error were:

0000 Disk read OK, but the format of its headers was probably wrong.
0001 Disk not recognised as being of correct type.
0002 Failure of find track.
0003 Failure to find sector.

The error message: 'SPAN ERROR. GET HELP' indicated the calibration check had failed, which could mean:

a. The 'SPAN DATA' was wrong or in the wrong position on the disk header.

b. The 'cal.' resistors were of the wrong value.

c. A zero offset had prevented null being obtained.

The AUX port baud rate buffer was at $10E0_{16}$, the TTY at $10C0_{16}$. These were set according to the following table:

9600 baud	-	2580_{16}
4800 baud	-	$12C0_{16}$
2400 baud	-	0960_{16}
1200 baud	-	0460_{16}
600 baud	-	0258_{16}
300 baud	-	$012C_{16}$

The Serial port format registers were at $10C2_{16}$ for the 'TTY' and $10E2_{16}$ for the 'AUX', and were set according to the following table:

7 BIT	No parity	1 stop	-	0082_{16}	
7 BIT	No parity	2 stop	-	0042_{16}	
7 BIT	Even parity	1 stop	-	$00A2_{16}$	
7 BIT	parity	2 stop	-	0062_{16}	
7 BIT	Odd parity	1 stop	-	$00B2_{16}$	
7 BIT	parity	2 stop	-	0072_{16}	
8 BIT	No parity	1 stop	-	0083_{16}	
8 BIT	No parity	2 stop	-	0043_{16}	- Default
8 BIT	Even parity	1 stop	-	$00A3_{16}$	
8 BIT	Even parity	2 stop	-	0063_{16}	
8 BIT	Odd parity	1 stop	-	$00B3_{16}$	
8 BIT	Odd parity	2 stop	-	0073_{16}	

Current program number is stores at 1010_{16}
Current flight number is stores at 1022_{16}
Current t/p number is stores at 1034_{16}

APPENDIX 5

Initializing the UARTS

In TDOS provision was made for operator selection of baud rates. The desired baud rate, expressed as a hexadecimal number, was loaded into location $10C0_{16}$ for the TTY port, and into $10E0_{16}$ for the AUX port. These ports operated independently, and very different baud rates could be employed. However both ports were returned to the default values whenever the system was reset.

The hexadecimal equivalents of the available baud rates were set out in Appendix 4.

The load programmer application software was so arranged that the baud rate settings for these two ports were changed to 300 baud for TTY and 9600 baud for AUX whenever this program was run. This application software could be modified to take advantage of the manual setting of baud rates in TDOS, but because this program was usually entered by resetting the system, it would result in trading one convenience for another.

With reference to source listings; to modify the application software it would be necessary to add a REF to SETSER, insert vectors for SCP and SAP, delete some redundant constants and change the 9901 and 9902 initializing routines as follows:

```
*     9902.1  9902.2        (Ref. 1)
      SETO @ UARTS
      BLWP @ SCP
      BLWP @ SAP

*     9901                  (Ref. 1)
      LI 12, $C0_{16}$
      SBO 0
      SBZ >F
      SBZ 0
      SBZ $1A_{16}$
      SBO 5
      SBO 7
      LI 12, $40_{16}$
      MOV 12, @ $1098_{16}$
      CLR 12
```

To be worthwhile it would also have been necessary to ensure baud rates remained valid on reset, and reverted to default values only on power-up.

Calculating the Baud Rate Word

Reference should be made to the section dealing with the 9902 in Ref. 1 page 8-160. This calculation was automatic in SETSER, but if baud rates other than 300 and 9600 were required with the current system it would be necessary to modify the application program. In particular constants TXT0 and/or TXT1 would have to be changed; values for these constants must be calculated. To illustrate the method used assume a baud rate of 300 was required:

$$\text{fRCV or fXMT} = 300 = \frac{10^6}{2xAxB}$$

The value for TXT was expressed as an 11 binary-bit word which defined both A and B. The most significant bit of the 11 binary-bit word determined whether A was to be 1 or 8. If this bit was 0 then A was 1. If the bit was 1 then A was 8. The remaining 10 bits were used to express the value of B. For 300 baud AxB had to equal 1667 and if A were 1 B must be 1667. Because 1667 could not be expressed by 10 binary bits A could not be 1. For this rate A had to be 8, ie. the 11th bit was a 1, and the remaining 10 bits represented the value 208 for B. This gave the 11 binary bit word as:

100,1101,0000 which in hexadecimal notation was $4D0_{16}$.

For a baud rate of 1200: AxB = 417.

Since 417 could be represented by 10 binary bits A could be 1 in this case.

417 was equivalent to 01,1010,0001

or, in hexadecimal notation, $1A1_{16}$.

In this case if A had been given a value of 8 B would have required to be 52; and the 11 binary bit word would have been: 100,0011,0100

or, in hexadecimal notation, 434_{16}.

APPENDIX 6

EXCURSION TIME OR CYCLING RATE

As mentioned in the report, command slew rate was determined by a hardware interval timer. Whenever this timer generated an interrupt the load programmer added/subtracted the next sequential increment to/from the current DAC command value. The timer generated an interrupt at the end of each timing interval (T); this interval being proportional to a value (R) loaded into its buffer. Increasing this interval increased the excursion time. The exact value for R to give a required cycling rate could be calculated but, because the dwell time at turning points was a variable, the actual period or rate achieved would only approximate the value required.

With the number of incremental steps to be added or subtracted to achieve a particular excursion known, and the excursion time (P) or frequency (f) specified, the value of R to be set in the timer was calculated from:

$$R = \frac{10^6}{42.66\ f(\frac{A-D}{B} + N)} \quad \text{for f in Hz}$$

or

$$R = \frac{10^6 P}{21.33(\frac{A-D}{B} + N)} \quad \text{for P in seconds}$$

To achieve the maximum possible excursion command, (from +10 volts to -10 volts) the input to the DAC had to be changed by 4096 units, and for any other excursion the number of units change required could be determined by expressing that excursion as a fraction of maximum excursion, ie.

$$A = \frac{\text{Excursion Required}}{\text{Maximum Excursion}} \times 4096$$

An excursion was built up from non-linear and linear increments; the smallest excursions requiring only non-linear increments. In any but the very small excursions the number of non-linear increments was fixed at the value N and these increments, when added together, contributed D units to the total excursion amplitude. The linear increments used to build up the excursion each contributed B units to the DAC input. In the current system N = 28, D = 156 and B = 11. these values could be changed only by changing the table and constants in firmware.

For the average excursion the number of linear increments required as $\frac{A-D}{B}$, and the total number of increments for the complete excursion was $\frac{A-D}{B} + N$.

For each increment required, irrespective of increment size, a fixed time T, determined by the interval timer had to elapse. This led to an expression for the excursion time of $(\frac{A-D}{B} + N)\ T$.

For the interval timer used in the stand alone controllers the buffer value R was decreased by 1 every 21.33 microseconds so that:

$$T = 21.33\ R \text{ microseconds.}$$

If P was specified in seconds the necessary value of R could be calculated:

$$R = \frac{10^6 P}{21.33(\frac{A-D}{B} + N)}$$

as previously stated.

Similarly where the cycling rate was specified in Hertz

$$P = \frac{1}{2f}$$

and

$$R = \frac{10^6}{42.66\ f(\frac{A-D}{B} + N)}$$

The nature of the interval time was such that the value of R calculated needed to be loaded into communcation-register-unit locations 1 to 14, and a 1 loaded into location 0. The 1 in location 0 was required to direct R into the appropriate clock register.

R expressed in binary was shifted left one position and a 1 added in the LSD position. The resulting number, expressed in hexadecimal was manually loaded into the appropriate buffer; $100A_{16}$ if there was a zero crossing, and $100C_{16}$ if not.

APPENDIX 7

A LIST OF SYSTEM MESSAGES

Generally the same messages were used for programmers, controllers and dual controllers. Where different messages were required for the different systems they have been labelled P,C or DC in the following list to indicate their application.

M1	C,DC	"The System will apply loads with amplitudes specified on floppy disks. System calibration is automatic and specimen calibration is possible. Loading may start from beginning of Program or at Turning Point reached in previous run".
M1	P	"The System will apply loads with amplitudes specified on floppy disks. Specimen calibration is possible. Loading may start from beginning of Program or at Turning Point reached in previous run".
M2		"RAM supply failed".
M3	C	"Header Error."
M3	DC	"Dwell Err. Check all Flt and T/P counters".
M3	P	"System not ready".
M4		"Reply with Y-CR for yes, N-CR for no".
M5		"Is Program correct?"
M6		"Wrong Disk. Replace and restart".
M7	C,DC	"Null or Span Error".
M7	P	"Header Error".
M8	C	"Hydraulics disabled".
M8	DC	"Hydraulics or Slaves disabled".
M9		"Is Data Logging required?"
M10		"DAU is off-line."
M11		"To start Program from first point type S; To go from point reached earlier type C."
M12		"S!!....OK to clear log?"
M13	C,DC	"Turn on Hydraulics".
M13	P	"Loading System ready?"
M14	C,DC	"Turning Point not found".
M14	P	"Turning Point not available. Get help".
M15		"Disk Read Error".
M16	C	"No null. Time-out. Shut-down active".
M16	DC	"Ack delay. Null/Ack error. Time-out. Shut-down active".
M16	P	"Time-out. Shut-down active".
M17		"Holding. At Turning Point. <T/P: Flt: Prgm: Time: Date:".

M18		"Type GO to start or continue testing. Type HD to hold at next Turning Point. Type WT to inhibit auto shut-down".
M19	C,DC	"Excursion failed to reach end-point value".
M19	P	"Excursion failed to reach end-point value. Shut down and get help".
M20		"Holding. At end of Flight. T/P: Flt: Prgm: Time: Date:".
M21		"DAU out. Check DAM Errors".
M22		"Flight Count Error".
M23		"At end of Program".
M24		Non printing message used to interrogate DAM.
M25		Non-printing message to DAM to read output of sensors.
M26		"Latch tripped. R5: ".
M27		Non-printing message to DAM to clear data buffers and stop.
M28	DC	"Header Error".

APPENDIX 8

A LIST OF SOFTWARE ROUTINES USED BY THE DATA ACQUISITION SYSTEM

1. Initialization including record search.
2. Console Receiver. Interrupt Handler.
3. Auxillary Receiver. Interrupt Handler.
4. Timer. Interrupt Handler.
5. Interrogate Command '$I_{\#}$'.
6. Input Message Command '<.....#'.
7. Sensor Reading Command 'R:.....;#'.
8. Terminate Current Run Command '$T_{\#}$'.
9. Terminate–Command Execution.
10. Echo Character.
11. Transfer Disk Buffer to Disk.
12. Fault Service.
13. Timer Service.
14. Multiplexer–Converter Controller.
15. Write on Disk.
16. Read from Disk.

APPENDIX 9

ACCEPTANCE TESTING

1. Control Modules (Load Programmer, Rig Monitor, Hydraulics Control and Servo Amplifier)

The two 5 volt supply cards were set to a current limit of 2 amps each, and one to 5.22 volts and the other to 5.20 volts. This was to prevent oscillation between supplies, and allow a margin for voltage drop in the cables. A check was made to ensure link 8 is installed, link 7 installed and link 9 removed.

On the 24 volt card (cct. 58241A2) link 8 was installed, link 9 removed and 7 installed. The Trim pots were set to 24.2 volts and 1.5 amps. This allowed for the surge current necessary to get the disk motor to speed quickly.

Once power supply operation had been verified other cards could be installed (one at a time, checking supplied were not shorted each time a new card was added).

First the switching logic card to the 'hydraulics control module', followed by the rig I/F card in the 'Load Program Module'. At this point supplies on the load programmer bus and the CPU card connector could be checked. Prior to inserting the CPU card in the load programmer, ST5 was linked to ST7 and ST8 linked to ST10. The monitor (TDOS) was installed in the sockets closest to the end of the card, M.S.PROM furthest from edge connector. The L.P. PROM installed in the IC14 position. The disk controller was inserted next as it controlled the DMA line. A terminal (set to 110,300,600 or 1200 baud, as selection is automatic) was then connected to the 'TEST' port. After pushing the 'RESET' button, waiting 2 seconds, and typing 'X' (upper case) the prompt '*' should be received. If not the following were checked:

1. Reset to pin 6 of 9900.

2. Clock to pin 8,9,25,28 of 9900.

3. -5 v to pin 1 of 9900.

4. Strobe to pin 20 of Monitor EPROM.

5. ASCII data reaching pin 4 of IC29 (74LS251).

6. ASCII '*' leaving pin 4 of IC30 (74LS259) in response to 'X'.

7. If DAM line is low.

Before the disk controller cards were adjusted a special cleaning disk was used to clean the drive's heads. IC12 pin 5 was then checked for a 350 nanosecond pulse when location 8000 was accessed (C13 and/or R4 being adjusted as necessary). Similarly IC5 pin 8 was checked for a 560-620 nanosecond pulse width while DMA was in progress (adjusting C14 and/or R2 if necessary).

On disk controller cards the centre trim pot was set to the half way position RV3 (closest to edge connector) to give 4 MHz at pin 16 of IC10 (WD1691). If this pot was near the limit of its travel adjustment of C14 to give a more central operating point was required (about 47 pF). RV1 (furthest from edge connector) was set fully clockwise, and advanced two turns at a time until reliable operation obtained. A check being made that link 2 and 16 were in place.

The clock on TTY card was adjusted by setting bit 74,75,76,77 and 79 to 1, bit 78 to 0, connecting a frequency meter to pin 9 of IC1 (MSM5832) and adjusting C13 for a reading of 1024.000 Hz. The 'SC' command could then be used to set actual time and data.

The rig I/F card was then installed and calibrated against an accurate Digital Volt Meter connected to the 'EXT. DVM' socket on the 'RIG MONITOR MODULE'. The 'AM8010' command was used to send 8000 to the DAC so RV2 could be adjusted to give a -10.000 v reading,. Then 7FFF was sent to RV1 could be set for a +9.9951 v reading; a check then being done that 0000 gave a 0.000 v reading. This procedure compensated for any errors in the differential input stage of this module.

This was a convenient point to adjust the trim pot at the rear of the meter in the 'rig monitor' unit so the panel meter gave the same reading as the 'accurate digital voltmeter'. The 'System test module' was then connected and RV3 and 4 adjusted for monitor null symmetry, and monitor null level (say 200 mv on command line) respectively.

2. Data Acquisition Module

This module was set up in the same manner as the Load Program Module except:

1. Its internal power supply had no adjustments.

2. It had no rig I/F card to worry about.

3. It used 'DO' PROM for IC14 on CPU card. (This put ordinary RAM in the 1000-17FF area used by the disk. Refer to Appendix 1).

4. It used DA EPROMS for IC15 and IC17.

5. Resistors R26,27,28,29 on 'Data Acquisition Card' (cct 58482 A1) were adjusted as necessary to give correct range (+4.9975 to -5.000 v) and zero. This was checked by connecting the DAM Test Box, setting it to a accurate DVM and using the 'CY' routine. (CY was a monitor routine causing the 8 channel ADC to read and print out reading values).

APPENDIX 10

SETTING UP THE TEST

1. Load Range

The system was first adjusted by mechanically unloading the cell to the condition defined as 'zero'. A resistor (or combination thereof) was then added to position R28 or R29 to give a zero reading on the DPM when it was switched to the 'control bridge' position (+/- a few millivolts). The monitor bridge was also adjusted along similar lines. Then the 'zero suppression' trimpot on the servo amplifier was adjusted to give zero on the DPM when in the 'feedback' position.

After ensuring DPM and DAC are correctly set up, a load representing about 80% of full scale was chosen. The 'command' was set to this value by, calculating the exact hex. number representing the load chosen, and setting the DAC to this value using 'AM8010' from the test port. Then a high accuracy decade resistance box was connected across one arm of the control bridge, as close to the cell as possible, and set to represent this load. Link 4 (on Rig Monitor), 'Feedback Gain', and 'Feedback Gain Trim' were then adjusted to give a reading of zero 'Error' on the DPM. The resistance box was then transferred to the 'Monitor' channel and reset as necessary. With the 'Command' unaltered, link 3 and RV2 (in Rig Monitor) were adjusted to position the residual signal in the centre of the null band.

2. Repeatability Checking Circuits

Shunt 'cal' resistor values were chosen to give about 60% full scale reading (this allows +/-40% variation in nominal zero load at which 'cal' was applied). The resistors were required to give equivalent loads (within 1%) in monitor and feedback circuits. An exact load equivalent was calculated for the feedback resistor, and used in the preparation of the load disk header. These feedback and monitor 'cal' resistors were placed in position R53 and R57 respectively (in Rig Monitor).

3. General

Before a real specimen was coupled to the loading system several checks needed to be made:

1. System polarity; tension in load cell = positive feedback voltage = jack travel in "compression" direction.
2. Travel limit micro switches and emergency stop switches all wired in and working.
3. Solenoid valves all operating properly (Remember: Failure to shut off supply when a fault occurs would be embarrassing).
4. Load limits on servo amp set correctly.
5. Servo Error limit on Servo Amp set appropriately (say 200 mv).
6. Servo gain not too high, say 100 (oscillating systems are also embarrassing).

7. Dither to about 50 mv pp for a start.

8. Span set appropriately (probably 100%).

9. Set Point to 500 or switched out.

10. Feedback null band set to small value.

11. Monitor null band set to larger value.

For more detailed information consult Reference 2 and 3.

APPENDIX 11

MOUNTING HARDWARE

Several factors had to be considered when mounting these systems:

1. Height, each 8 channels of data Acquisition required 3E, each control channel 9E plus about 3E per additional channel for intersystem modules. NB: E = 44.45 mm = 1.75 inches.

2. Depth, from the front mounting flange the units required a minimum of 450 mm. This however did not allow air flow through the cooling fans, so if this dimension was used a ventilated rear door would be needed. The vent should be equipped with some kind of effective filter.

3. Accessibility was a major concern with the Rig Monitor as it contained many internal adjustments. Hence telescopic sliders were recommended for this module. However as the normal width of a cabinet was 450 mm and the width of the type 115 module 443 mm, special rear sections for the frame needed to be fabricated (see Drawing 60096A1), and the sliders still had a maximum thickness of 10 mm each (3.5 mm each side standard plus 6.5 mm for modified rear vertical sections). The maximum height was limited to 66 mm by the side rails. These requirements were met by the ELMA 65-001-03 rails (457.2 mm long).

4. Cooling did not present a problem if only one basic system was in the rack. However if two or more channels were mounted in the one cabinet it was recommended forced draught be provided by one or more fans. The positioning of these fans, had to take into account the inability of air to flow vertically through the rig monitor module.

APPENDIX 12

REVISIONS

From experience gained with the prototype since the original order was placed some improvements have been made.

1. Span range on rig interface card needed increasing (R4 changed to 11K).

2. DAM reset switch was replaced by a EAO 19-139.0 switch, 19-932.0 lens and 99-943.02 LED (the cathode being taken to SK2-J on the CPU I/F card via 390 ohms, the anode to +5 v PL1-EE).
NB: Hole needed enlarging to 8.0 mm. This gave a visual indication of an error condition.

3. A similar device was added to the Rig Monitor module in place of the 'Dwell' LED, with one side taken to digital common and the other to the base of TR2. This allowed dwell to be manually inhibited to perform valve balance.

4. To improve long term accuracy of rig monitor R61 was replaced with a high stability 1 M, and RV1 with a fixed high stability 110 K (range adjustment available at rear of meter).

5. Problems were encountered achieving certain gain settings, these were overcome by:

   a. Changing R14 on the Servo Amplifier, Input Amplifier card (51 903 A2) from 4K7 to 2K7.

   b. Replacing R63,64,65 and RV2 (10K,10K,4K7 and 10K) on the Rig Monitor (58 387 A1) with 47K,47K,9K1 and 50K respectively.

6. C15 on disk controller card B (58 219 A1) could be reduced from 330 nF to 33 nF to improve PLL performance.

7. Later versions of the software have been released incorporating checks on the reading of disk blocks to ensure spectrum is read accurately.

8. A later version of the Load Programmer software incorporates:

   Inspect command to stop at given prog/flight/T/P.

   Z to suppress end of flight logs.

   Photo command to trigger a camera.

This version needed a special TDOS with vectors relocated because it overran the old vector area.

DISTRIBUTION

AUSTRALIA

Department of Defence

Defence Central
Chief Defence Scientist
Assist Chief Defence Scientist, Operations (shared copy)
Assist Chief Defence Scientist, Policy (shared copy)
Director, Departmental Publications
Counsellor, Defence Science London (Doc Data Sheet Only)
Counsellor, Defence Science Washington (Doc Data Sheet Only)
S.A. to Thailand Military R and D Centre (Doc Data Sheet Only)
S.A. to the DRC (Kuala Lumpur) (Doc Data Sheet Only)
OIC TRS, Defence Central Library
Document Exchange Centre, DISB (18 copies)
Joint Intelligence Organisation
Librarian H Block, Victoria Barracks, Melbourne
Director General - Army Development (NSO) (4 copies)

Aeronautical Research Laboratory
Director
Library
Divisional File - Aircraft Structures
Authors: E.S. Moody
I. Powlesland
A. Patterson
P. Ferrarotto
D. Smith
L. Conder
C. Ludowyk

Materials Research Laboratory
Director/Library

Defence Science & Technology Organisation - Salisbury
Library

Navy Office
Navy Scientific Adviser (3 copies Doc Data sheet only)

Army Office
Scientific Adviser - Army (Doc Data sheet only)
Engineering Development Establishment, Library

Air Force Office
Air Force Scientific Adviser (Doc Data sheet only)
Aircraft Research and Development Unit
Library
Engineering Division Library

Statutory and State Authorities and Industry

Aero-Space Technologies Australia P/L - Manager/Librarian (2 copies)
BHP, Melbourne Research Laboratories
Hawker de Havilland Aust. Pty Ltd, Victoria, Library
Hawker de Havilland Aust. Pty Ltd, Bankstown, Library

Universities and Colleges

RMIT
Library
Aeronautical Engineering Department

SPARES (10 copies)
TOTAL (59 copies)

AL 149
REVISED DECEMBER 87

DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER AR-005-525	1b. ESTABLISHMENT NUMBER ARL-STRUC-TM-488	2. DOCUMENT DATE JUNE 1988	3. TASK NUMBER DST 82/007

4. TITLE
SINGLE CHANNEL TEST CONTROLLERS

5. SECURITY CLASSIFICATION
(PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONFIDENTIAL (C), RESTRICTED (R), UNCLASSIFIED (U).)

U	U	U
DOCUMENT	TITLE	ABSTRACT

6. No. PAGES
51

7. No. REFS.
4

8. AUTHOR(S)
I. POWLESLAND
E.S. MOODY

9. DOWNGRADING/DELIMITING INSTRUCTIONS

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY
P.O. BOX 4331, MELBOURNE VIC. 3001

11. OFFICE/POSITION RESPONSIBLE FOR

SPONSOR ______________
SECURITY ______________
DOWNGRADING ______________
APPROVAL ______________

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT) Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH ASDIS, DEFENCE INFORMATION SERVICES BRANCH, DEPARTMENT OF DEFENCE, CAMPBELL PARK, CANBERRA, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO.....
No limitation

13b. CITATION FOR OTHER PURPOSES (IE. CASUAL ANNOUNCEMENT) MAY BE □ UNRESTRICTED OR □ AS FOR 13a.

14. DESCRIPTORS
Aircraft structures
Aerodynamic loads
Test equipment
Controllers

15. DRDA SUBJECT CATEGORIES
0051F

16. ABSTRACT

The stand-alone controllers in single and coupled-channel forms are described. Details of hardware and software are given, and notes on the associated data acquisition system and test boxes are included. A number of appendices provide data for preparation and reading of floppy disks, for cycling and communication rate adjustments and detailed information on other aspects of these units.

[illegible]

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT.)

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER AIRCRAFT STRUCTURES TECHNICAL MEMORANDUM 488	19. COST CODE 22 5760	20. TYPE OF REPORT AND PERIOD COVERED

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)