

NLR MP 83040 U

Bibliotheek TU Delft  
Faculteit der Luchtvaart- en Ruimtevaarttechniek  
Kluyverweg 1  
2629 HS Delft

# NATIONAAL LUCHT- EN RUIMTEVAARTLABORATORIUM

NATIONAL AEROSPACE LABORATORY NLR

THE NETHERLANDS

NLR MP 83040 U

## VERIFICATION AND VALIDATION OF IRAS ON-BOARD SOFTWARE

BY

R.J. NICOLAI





NLR MP 83040 U


VERIFICATION AND VALIDATION OF IRAS  
ON-BOARD SOFTWARE

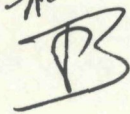
by  
R.J. Nicolai

The contents of this paper is to be presented on the ESA/ESTEC Software Engineering Seminar to be held 11 - 14 October 1983 in Noordwijk, The Netherlands

(7 pages in total)

Division: Informatics

Prepared: RJN/ 

Approved: 

Completed : 23-VI-1983

Ordernumber: 51.306

Typ. : MG



VERIFICATION AND VALIDATION OF IRAS ON BOARD SOFTWARE

R.J. Nicolai

National Aerospace Laboratory Amsterdam

ABSTRACT

The infrared Astronomical Satellite IRAS is equipped with an on-board computer system that takes care of vital satellite functions as attitude control, ground dialogue and execution of the observation programs. Therefore an important purpose of the IRAS system tests was the Verification and Validation of the on-board software. The test configuration was largely determined by the attitude control system. Provision of representative inputs for the attitude control system tests, required orbital simulation. The attitude control functions resident in ROM were exercised extensively in closed loop (satellite dynamics model in Ground Check-out computer), providing a flexible environment for combined "black and white box" testing. The less critical but more accurate RAM software had to be exercised in open loop (attitude sensor stimuli precomputed on a mainframe). The large amount of RAM software permitted only functional ("black box") testing. The closed loop method was confirmed to be the most desirable.

Keywords: Verification, Validation, Reliability, System test, on-board computer, On-board software, IRAS.

CONTENTS

- 1 INTRODUCTION
- 2 IRAS CONFIGURATION
- 3 SYSTEM TEST CONFIGURATION
- 4 SYSTEM TEST PREPARATION
- 5 ROM SOFTWARE VERIFICATION AND VALIDATION
- 6 RAM SOFTWARE VERIFICATION AND VALIDATION
- 7 DEBUGGING
- 8 CONCLUSION
- 9 ABBREVIATIONS

LIST OF FIGURES

1. System test configuration
2. ROM software test configuration
3. RAM software test configuration

1. INTRODUCTION

The Infra Red Astronomical Satellite IRAS was launched January 26, 1983 for a mission that will last approximately until January 1984. Its purpose is to produce an all-sky catalogue in the infrared waveband and to make special observations of selected sources. After 5 month life time the satellite is still performing extremely well.

IRAS is equipped with an on-board computer system that takes care of satellite functions as attitude control, ground dialogue and execution of the observation programs. Therefore an important purpose of the IRAS system tests is the Verification\* and Validation\* of the on-board software. Especially the correctness of the software in ROM, taking care of vital functions like ground communication and safeguarding the delicate experiments from excessive heatinput (sun, earth), is of major importance.

Exercising the attitude control software during Verification and Validation both in ROM and RAM required orbital simulation for provision of representative inputs. The ROM attitude control software could be exercised in closed loop thanks to its coarse character, rendering a flexible environment for extensive tests. Due to its accuracy requirements, the RAM attitude control software had to be exercised in open loop with stimuli precomputed on a mainframe computer.

The interaction of the on-board software with the other satellite subsystems is more straight forward. During system tests the interfaces with the power-system and the telemetry and command system could therefore be implicitly validated through the supporting testequipment (power) and the use of the GCE (communication). Interfaces with experiments and recorders consisted of fixed data formats. Consequently the testconfiguration was largely determined by the attitude control system. This paper therefore concentrates on the system level tests of the software of the attitude control system.

The system tests applied to IRAS were the result of a cooperative effort of people from:

Fokker B.V., Amsterdam  
Hollandse Signaalapparaten B.V., Hengelo  
National Aerospace Laboratory NLR,  
Amsterdam



During these tests the NLR contribution was specially directed to the provision of test tools for and testing of the on-board software.

*Not*

\*Verification is defined as the process of checking that the software perform according to specification and user description.  
Validation is defined as the process of checking that the software comply with the user objectives. (The specifications may not correctly reflect the user objectives.)

## 2. IRAS CONFIGURATION

IRAS consists of two main parts:

- 1) The infrared telescope with related electronics, enclosed in a liquid-helium dewar, 270°C below zero, so extremely vulnerable for heat-input.
- 2) The spacecraft itself, containing the subsystems for the following functions:
  - ground communication
  - attitude control
  - experiment control and data handling

The IRAS system configuration can be characterized as follows: Its heart is formed by two on-board computers (one is cold standby), using four memory blocks. The computers communicate via their own bus system with the subsystems:

- Attitude Control
- Power
- Telemetry and Command
- Experiments
- On-board Recorders.

The ACS sensors are:

- fine sunsensor (2 axes)
- 4 gyros (2 redundant ones for z-axis)
- horizon sensor
- 6 coarse sunsensors
- magnetometer

The ACS actuators are:

- 3 reaction wheels
- 3 magnetic coils

The memory is divided in ROM and RAM. The software residing in ROM (6k bytes) takes care of vital functions:

- ground communication (commanding, telemetry)
- coarse attitude control (accuracy  $\pm 5$  degrees), with the objective to maintain a safe attitude (no heat input to the delicate experiments).

The operational software resides in RAM (20k bytes).

Each twelve hours a new observation program, the Satellite Operations Plan (SOP), is loaded in RAM. Directed by this SOP, the following functions are performed:

- experiment control and data acquisition
- fine attitude control (accuracy 10 arcsec)

The ROM software cannot be corrupted (reliability), while the RAM software can be reloaded from the groundstation, accommodating possible changing operational requirements (flexibility).

The operating systems (both in ROM and RAM) are driven by a 256 Hz clock interrupt, providing 256 timeslots for activation of application software (attitude control, data handling). All software was written in assembler.

## 3. SYSTEM TEST CONFIGURATION (fig. 1)

The system level tests were conducted with the aid of ESA Overall Checkout Equipment adapted for IRAS, called Ground Checkout Equipment (GCE). IRAS was

the first satellite containing a reprogrammable OBC using this equipment. Consequently a major adaption was the introduction of functions dealing with OBC programs (filing, loading, dumping and comparing).

The GCE interfaces with the satellite via:

- 1 radiofrequency interface, which is also the operational interface (commanding, telemetry)
- 2 umbilical connectors providing power and video frequency communication
- 1 testconnector for direct access to the OBC bus
- a stimuli rack for stimulation (electrically or physically) of the sensors of the attitude control subsystem.

The stimuli were used for simulation of the orbital environment, while the testconnector was mainly used for software debugging. The experiment detectors could be stimulated by means of internal stimuli in the telescope.

The GCE computer directed the testing process through automated test sequences generating stimuli and commands, while monitoring the telemetry.

## 4. SYSTEM TEST PREPARATION

Before the actual system tests were started all interfaces of the OBC's with the units were verified (hardware, software, timing), the so-called hardware-software integration tests. The operational software was not very fit for this purpose since it expects to deal with a completely integrated system. For this reason it was partly replaced by testsoftware, which ran under the original RAM operating system and used the original input/output routines. This testsoftware was organised in "jobs" related with the various units and subsystems, and could be inter-actively controlled from the GCE, using ASCII messages communicated via the OBC test connector. In this way attention could be focussed on a specific interface, while possible interference between units could be investigated in a controlled manner.

## 5. ROM SOFTWARE VERIFICATION AND VALIDATION

On the subsystem level the ROM software was extensively tested on the commercial version of the OBC, the Philips P856S. On system level the focus could be on system functions and on the interface aspects with the units, whereby all attitude modes were exercised.

For simulation of the environment a closed loop method was selected which had the following advantages:

- changing test requirements could be accommodated without additional (mainframe) simulations.
- test results could easily be interpreted (interaction of system and test environment result in "real" attitude and rotational velocity).

Fig. 2 shows the set up: A satellite dynamics model in the GCE computer uses as inputs the reaction wheel velocities and the magnetic coil setpoints as present in the telemetry. The resulting attitude and rotational velocity were converted into electrical or physical stimuli for the attitude sensors using the relevant transfer functions. The ROM attitude control software used the sensor information for computation of setpoints for reaction wheels and magnetic coils.

The limitations for the closed loop method were:

- 2 second delay in loop due to data exchange and computation
- one second update cycle of the sensors (setpoint calculation two times per second)



- integration of dynamics equations one time per second (computing power of testcomputer was limited).

Due to the "coarse" character of all ROM algorithms (2 degrees resolution), none of these limitations were significant as proven by mainframe simulations. The module governing the attitude mode was tested with special regard to its internal structure, since this module had many interfaces with the hardware (status, health bits), which could not be verified elsewhere. All conditional branches were exercised, at least once in either way. For the complete ROM software this was not attempted (prohibitive on system level).

Seven test runs were conducted with different initial conditions. Manipulation of the environment (unit health, eclipse) at specific moments, resulted in the intended paths in the control module being followed. In this way all modes and the transitions between them were exercised.

In order to shorten the overall test duration, timers and counters in the ROM work area (in RAM) were changed by means of memory loads. Essentially this is a combination of "black box" testing (satellite dynamics model) and "white box" testing (path testing).

To validate the satellite dynamics model itself, the results were partly compared against simulations carried out on a mainframe computer.

A total number of 10 errors was found. None of them was the result of deviation from the specification (verification). They were either the result of unit malfunction, omissions in the specifications or wrong assumptions about the functioning of units and/or interfaces, which stresses the importance of Validation.

Two examples of these errors are:

- The horizon sensor output could be acquired by the software in two ways, directly or via the telemetry and command unit. In the latter case the "accept" status indicating the result of the read out instruction was OK by default. So, if the horizon sensor was switched off, this was not recognised by the software. In that case the required backup mode was not selected, resulting in the satellite rotating upside down. An extra check for the horizon sensor power status cured this problem.
- After switching to the redundant OBC because of an attitude out of limits status, this OBC was continually re-initialised due to a re-arming of the relevant circuit by the ROM initialisation routine. Deletion of the blocking command for OBC switching cured this problem.

#### 6. RAM SOFTWARE VERIFICATION AND VALIDATION

The simulation of the environment for Verification and Validation of the RAM software had to take place in open loop. The preferred closed loop method would have resulted in an unrealistic large limit-cycle and inaccurate sensor values (arcsec range) due to the limitations mentioned in the previous section.

The open looptest procedure is illustrated in figure 3: Sensorstimuli, precomputed on a mainframe computer as a result of a chronological list of attitude setpoints, were applied to the sensors by the GCE. The onboard software, driven by the same list of attitude setpoints as present in the SOP, computes the actuator setpoints, which were recorded by the GCE (stripcharts).

These setpoints were visually compared against the results of the mainframe simulations. Longterm accuracy of the attitude was maintained through accumu-

lative comparison of actual with required gyro output.

The star sensor used for fine attitude calibrations, could not be directly stimulated (starsensor is part of the experiment). Star passages were simulated by flashes from the internal visual simulators, commanded by the SOP, providing the synchronisation with the attitude setpoint generation. The timing of the flashes was fixed (msec resolution), resulting in reproducible attitude errors, which were accounted for in the mainframe simulations.

Control of the other subsystems (experiments, recorders) was also governed by the SOP. The resulting testdata was processed off-line (checking data formats etc.).

During system test of the RAM software (20k) only functional testing was performed. All main and backup modes and most of the transitions between them were exercised, without regard to the internal structure of the software (black box testing).

Apart from the fact that it is impractical to apply more rigorous tests at the system level for this amount of code because of time and money constraints, this method could be afforded because:

- possible remaining errors can not be disastrous (ROM functions take over)
- after launch errors can be corrected (the RAM software can be completely reprogrammed and reloaded)
- performance related aspects of the attitude control software were verified on a single axis Attitude Control Model by different people at a different site.

A total number of about 120 errors (of which 30 errors in the specification) was found in the course of a three years period, whereby the RAM software was changed up to the last weeks before launch o.a. because of launch delay. The majority of these errors were interface errors between the software modules. The omission of a software integration test at the subsystem level in the P856 due to shortage of memory was probably responsible for this effect. In the last phase of the project the largely automated and reproducible RAM tests had the character of regression tests, whereby it was verified that newly introduced changes did not affect the original performance.

An example of a late change of the RAM software was the increase of the update interval of the magnetic coils, whose outputs disturbed the experiment detector outputs. The software change and verification was a matter of days, while a hardware change would have required disintegration of satellite and the relevant unit.

The open loop method lacked flexibility. This was worsened by the large geographical distance between the mainframe computer and the satellite test site (e.g. Netherlands - USA). Changes in the software in later stages of the project, had to be verified at the subsystem level. Sometimes the closed loop method was a useful alternative if the related limitations were of no consequence.

#### 7. DEBUGGING

If it was not immediately clear which part of the software was responsible for a detected error, the visibility of the processes in the on-board software had to be increased. This was accomplished by loading a separate module in the on-board computer which ran under the RAM operating system. This module sampled designated addresses at selectable instants (one of the 256 timeslots of the 1 second



cycle), their contents being transmitted via the OBC testconnector to the GCE, where they were printed for off-line examination. Due to the 80 % idle time of the central processing unit this module did not affect the performance of the actual on-board software.

TSY - telescope  
UCR1,2 - umbilical connector  
V and V - verification and validation

Even during in Orbit Checkout the GCE, in conjunction with the Spacecraft Electrical Model, proved to be valuable for validation of software changes. An occasional malfunction of the fine sun sensor required an adaption of the on-board software which was first validated by means of above-mentioned system test configuration.

#### 8. CONCLUSION

Of the IRAS on-board software the verification and validation of the attitude control software required most of the effort. This stems mainly from the fact that the real environment (in orbit) had to be simulated.

Simulation of the environment for the programs, resident in ROM, could be done in closed loop. The inherent inaccuracy due to a.o. limited computational power and loop delay could be afforded thanks to the sensor resolution ( $2^0$ ) used in these algorithms. The errors found in relation with the ROM software all resulted from incorrect or insufficient external specifications, which stresses the importance of validation.

The environment of the software resident in RAM had to be simulated open loop due to the accuracy requirements. This open loop method lacked flexibility. Most errors found were the result of incorrect interfacing between software modules, which stresses the importance of interface testing at the subsystem level.

Experience gained with both open and closed loop environmental simulation confirmed that the latter method is very desirable. It however requires a fast and powerful computer/language to be part of the GCE, and direct access (without time delay) to the on-board computed actuator setpoints and the sensor stimulators.

The IRAS mission has lasted for 5 months now and is still very successful. Only 3 minor errors were found and corrected in the RAM software during operation. Above that a number of software changes has been implemented facilitating operational use.

#### 9. ABBREVIATIONS

ACE - attitude control electronics  
CSS - coarse solar sensors  
DAX - Dutch additional experiment  
FSS - fine sun sensor  
GCE - ground checkout equipment  
GYR - gyro's  
HSE - horisonsensor  
HSF - high speed frame  
IRAS - Infra Red Astronomical Satellite  
LSF - low speed frame  
MCL - magnetic coils  
MGM - magnetometer  
OBC - on board computer  
OBS - on board software  
RAM - random access memory  
RF - radio frequency  
ROM - read only memory  
RWL - reaction wheels  
SOP - satellite operations plan  
SSE - star sensor  
TCR - test connector



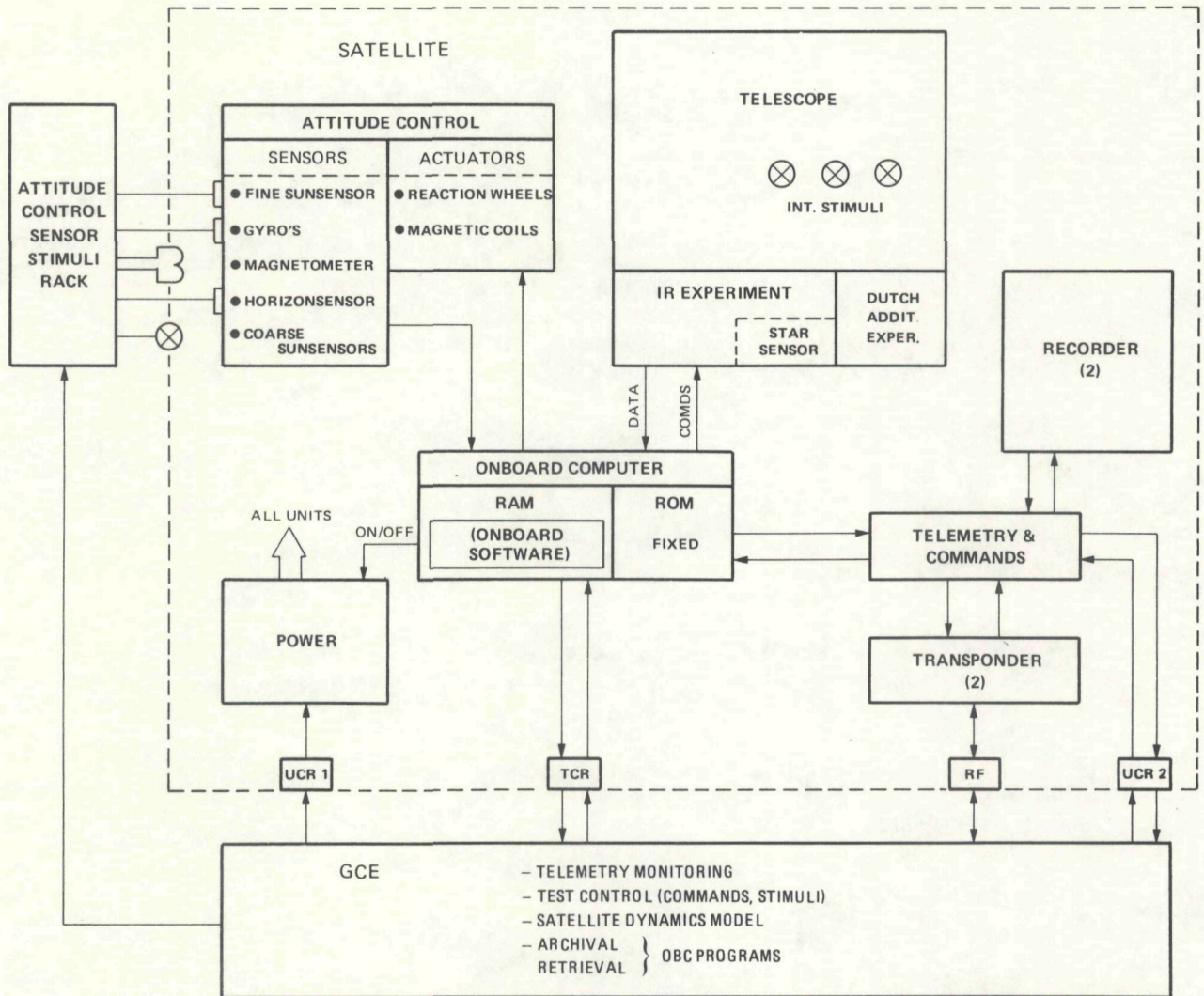


Fig. 1 System test configuration



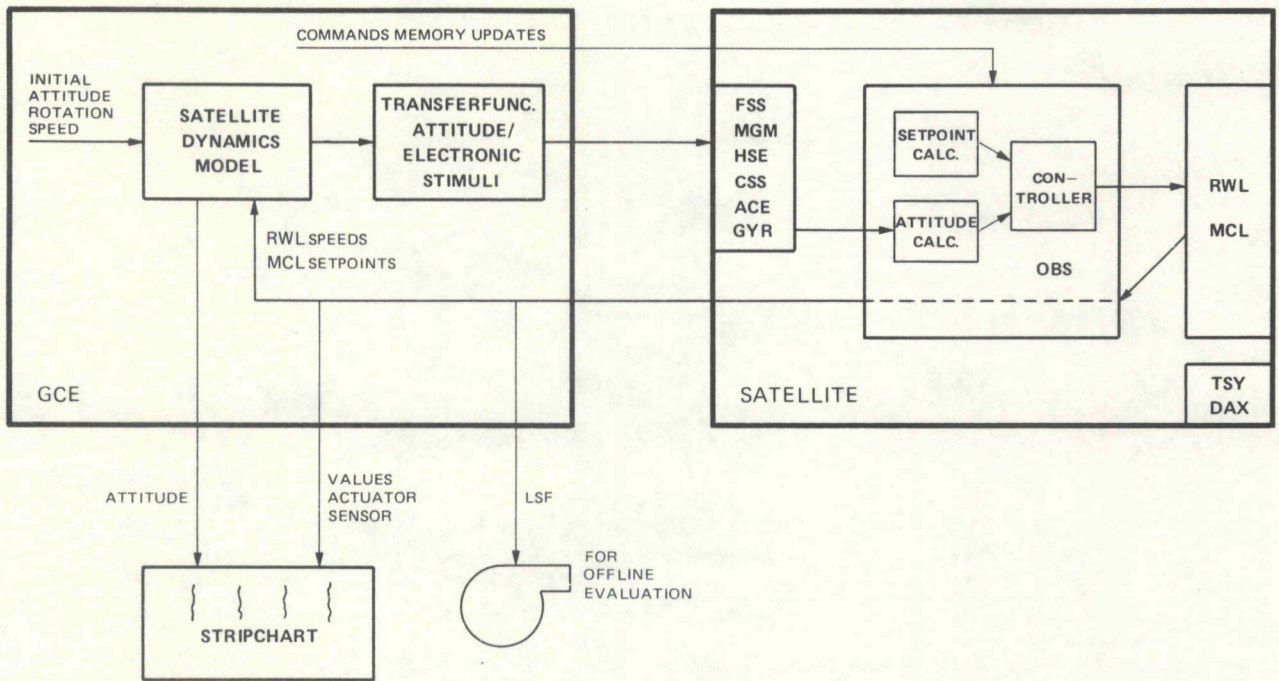


Fig. 2 ROM software test configuration

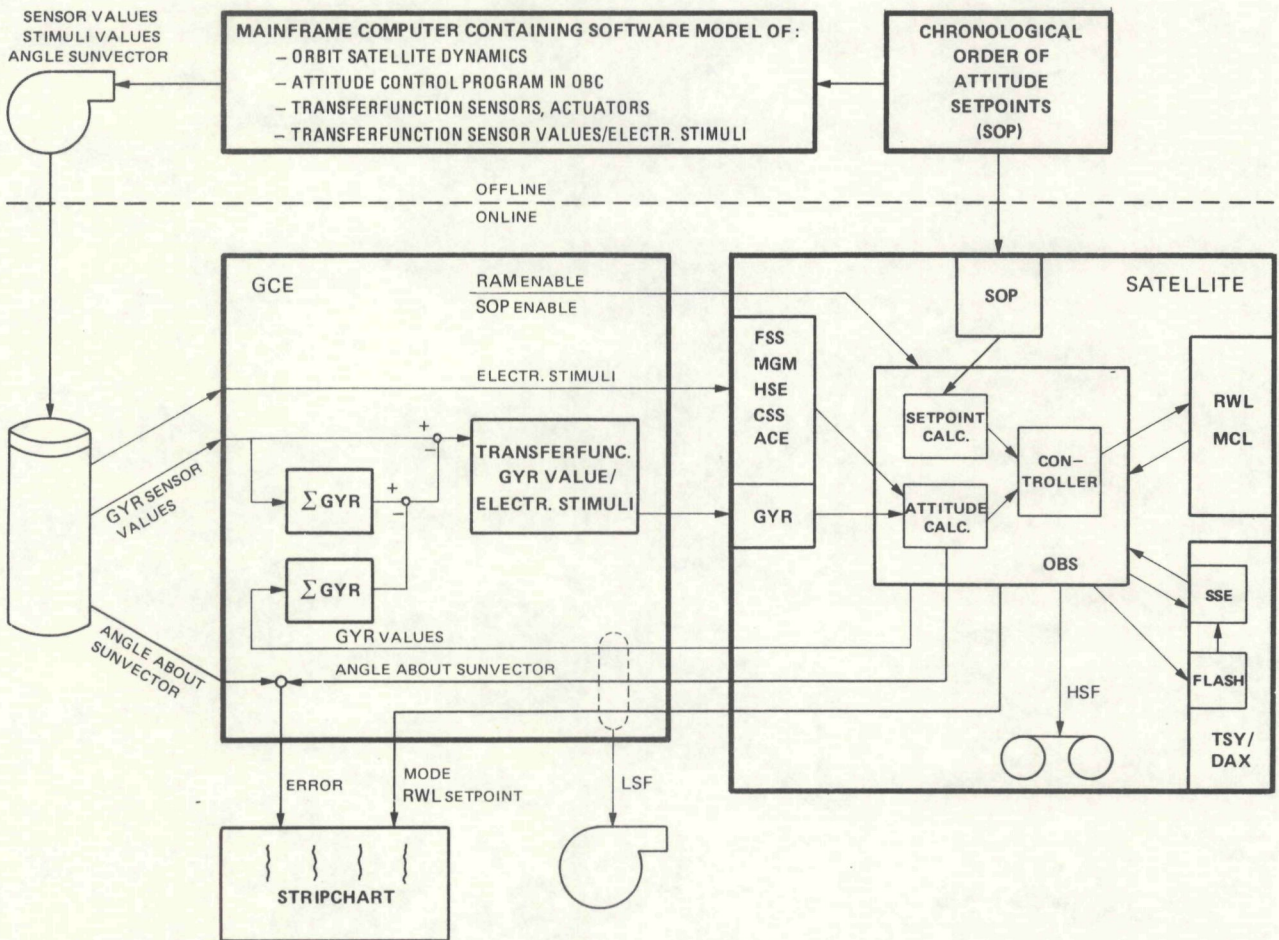


Fig. 3 RAM software test configuration