International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 02 | May-2015 www.irjet.net p-ISSN: 2395-0072

Simulation and design of a fault tolerant autonomy of a spacecraft using leon-3 processor

[1]K V S Krishna Chaitanya,Computer Science, MSRIT, karnataka,India

[2] Udit Khiraiya,Computer Science, MSRIT, karnataka,India

[3] Vinit Kumar R B,Computer Science, MSRIT, karnataka,India

[4] Nitin S Mali,Computer Science, MSRIT, karnataka,India

[5] Jagadish S Kallimani,Computer Science, MSRIT, karnataka,India

[6] Prashant Kulshrestha,ISRO, karnataka,India

---***---

Abstract - ***Spacecraft perform a variety of useful tasks that help us in our day to day lives. Be it watching a live feed of a boxing match in the other side of the globe or giving directions to a location. Without the advent of satellites humans would be lost in the literal and metaphorical sense. Once launched, satellites need to function in space without interruption for seven to fifteen years without any physical maintenance, until decommissioned. In case of failure most spacecraft have redundant systems to serve as back-up they also greatly depend on human assistance from an available ground station. Having mentioned that, it takes hours or sometimes even days for a satellite to communicate with its corresponding ground station. What happens if we have a scenario where in a fault in a satellite needs to be dealt with immediately? There is, therefore, a need for a fault tolerant system within a satellite that detects a fault, isolates it, and rectifies it and bringing the satellite back to its original safe state without any assistance from humans whatsoever.***

Systems on-board usually use LEON-3 processors, which are coded a programming language known as ADA. This paper describes an efficient scheme to design and simulate a fault tolerant system on a LEON-3 processor using tools such as a cross compiler, eclipse and TSIM, a combination of which enables us to code in a simple language such as C .Also this paper proposes a basic architecture of a fault tolerant satellite.

Key Words: *Satellite, Failure, Fault Tolerant System, Safe State, LEON-3, ADA.*

1. INTRODUCTION

Man has made giant strides into space exploration. Contemporary missions involve launching satellites into space to primarily facilitate communication, mapping of the earth and imaging. Satellites have also been used by the defence forces. A satellite consists of many components. Each of these components is very complex entities made up of tens of thousands of smaller components that interact with each other to achieve a set of objectives. Occasionally, unexpected behaviour or faults are generated. These faults must be appropriately addressed to ensure seamless and continued functionality of the spacecraft. In our modern society, spacecraft play many important roles, domestic and international telecommunication broadcasting, weather forecasting, remote sensing etc. And their services have become essential in our day-to-day life. Hence there is a growing need to provide uninterrupted operation of spacecraft over very long periods of 10 to 15 years. However, despite various efforts to improve reliability of a system through 'fault avoidance' techniques such as improvements in design and fabrication, use of high reliability and burnt –in or screened components and elaborate and intensive testing, failures do occur in various subsystems during their long operational life. Failure of even one of the components/subsystems might lead to the malfunction of entire control system which, in turn, might result in aborting the mission.

2. MOTIVATION

Most of the earlier and current spacecraft control systems generally have redundant units/subsystems to achieve required reliability and to mitigate mission critical 'single-

point-failures', they are, however greatly dependent on ground support for decision making and management of redundant unit. Diagnosis and adaption to faults is carried out by mission/subsystem specialists on ground through careful analysis performance and status information telemetered (transmitted) to the ground. Depending on the nature of faults remedial actions are taken through telecommands to affect recovery and to bring the spacecraft back to normal operations.
Further, this approach suffers from the following limitations:
Due to inherent delays in taking corrective actions from ground, failures such as free flow of fuel through thrusters and the speed of the reaction/momentum wheel goes beyond its absolute maximum limits, might lead to catastrophic effects.
In a crisis like a natural calamity, or an external threat, when continued spacecraft operational support would be required more than ever, ground contact and control could be interrupted for long periods.
Thus there is a need to design and incorporate a fault-tolerant system that performs its functions autonomously despite failures. This can be achieved using redundant subsystems and detecting behaviour of subsystems on board the spacecraft, with full autonomy to switch automatically to redundant units in case of failures. This approach also simplifies ground station operations significantly[1].
Shifting focus onto why we're using the Cross compiler, Tsim and Eclipse combo to implement a simulation of the above, it's mainly done so that one can get a feel of working with the expensive Leon-3 processor by simply using the simulator (Tsim). Also, one can develop programs on it (in our case, fault tolerant satellite system) using simple C instead of ADA (GCC Cross compiler + Eclipse). Hence the problem definition is to simulate and design of a fault tolerant autonomy of a spacecraft using leon-3 processor.

3. LITERATURE SURVEY

For autonomous spacecraft, anomaly detection and fault diagnosis systems play a very important role. To meet the requirement of real-time diagnosis performance for the fault diagnosis system of an autonomous spacecraft. A multiple observer based scheme is proposed jointly with an online constrained allocation algorithm to detect, isolate and accommodate a single fault affecting the propulsion system of an autonomous spacecraft.
Therefore, advanced Fault Detection and Isolation (FDI) approaches should be specifically developed to safely conjugate the necessary robustness/stability of the spacecraft control, trajectory dynamics and the vehicle nominal performance
The aim is to develop an algorithm which can quickly detect, isolate, and accommodate single thrusters fault in a simple manner and is easily implementable for a real spacecraft mission. As soon as a fault is declared to be faulty by the FDI unit, the associated (faulty) unit is closed. This fault accommodation strategy is achieved by fault autonomous.
A satellite consists of a computer onboard that manages and coordinates all the sub systems of the spacecraft. At a high level, the structure of a space craft could be represented as in the fig1. Sensors, actuators tele-command and telemetry form the major sub systems. The sensors include devices that obtain data from the environment such as a camera and other payloads. The actuators include torquers, thrusters and wheels. The spacecraft can be oriented and directed using these devices. Tele-command includes all the signals that are sent to the spacecraft from the ground. Telemetry includes all the data (other than application data) sent by the spacecraft to the ground. Application data is the useful data from the satellite that can be used for a variety of purposes.
Currently, only few kinds of autonomous fault monitoring tools system are available which provide fault detection methods. It consists of different modules for each function and these modules try and fix the faults.
These faults vary and hence it depends on the faults which lead the necessary action to be taken against it.
We are currently using sparc bare-c cross compiler and tsim simulator to work. This interface is compatible to work on eclipse Helios. The processor that is being used on spacecraft is leon-3.the development of the code is in c-language. The leon-3 processor is responsible for the steps taken in the spacecraft.
This software targets the debugging of applications and making the detection easier. The system and its parts have been discussed below in this diagram. The sensor, actuator, telecommand and telemetry make way to process the spacecraft and function accordingly. The fault autonomous part is separately handled by the processor and it assigns the task with the scheduling algorithms.

4. DESCRIPTION

A satellite consists of a computer onboard that manages and coordinates all the sub systems of the spacecraft. At a high level, the structure of a space craft could be represented as in the figure 1. Sensors, actuators tele-command and telemetry. The satellite constantly sends data back to the ground about its health. When anomalies are detected, signals are sent to the spacecraft to initiate recovery procedures. Due to the rarity of such faults, there is a need to automate fault correction so that a large extent of faults can be corrected by the spacecraft itself. This approach also reduces the data sent to the ground stations thereby saving bandwidth which can then be used

by applications (Imaging, Defence, Remote Sensing and Communications)[2].
The following three steps are a part of the fault tolerance automation:

- Detection
- Isolation
- Reconfiguration

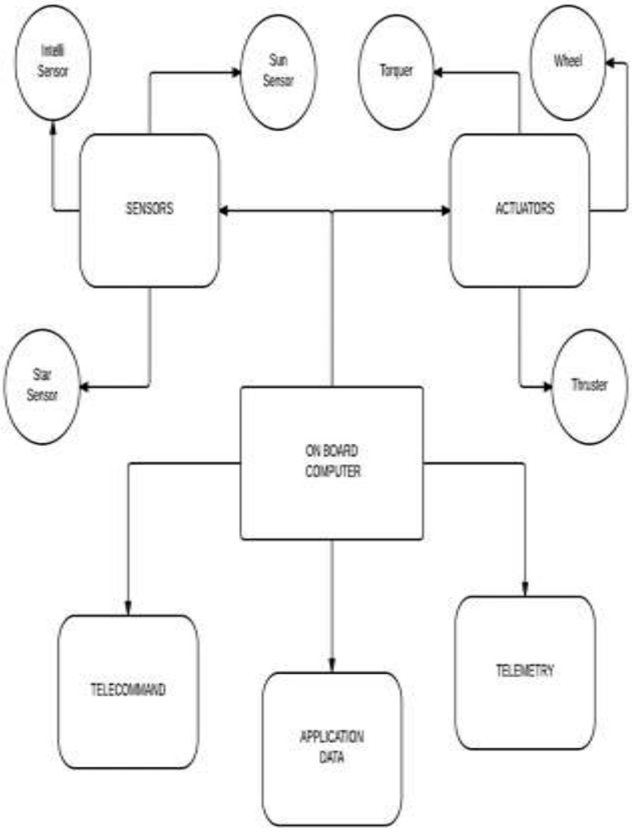


Fig -1: Structure of a space craft

Detection involves fault definition: The states of the spacecraft when a fault is said to have occurred. The state of the spacecraft is constantly monitored by the on board computer to determine if such a state has occurred. This can be achieved by each of the sub systems sending a periodic signal (a heartbeat) to the computer on board. The second step is isolation. The faulty component is identified and then isolated. This ensures that the faulty components cannot generate any more faults. The last step is reconfiguration. Most components of a spacecraft have a redundancy to combat failure. When faults are detected and it is established that these are due to component failures, the failed components are isolated and the paths to these components are redirected to the redundant components on board. This results in seamless and continued availability of the space craft and prolongs the life of the spacecraft[3].
A cross compiler is used to compile the code to the target platform (LEON3). The processor is simulated using tools (TSIM LEON3 simulator).faults are introduced into the spacecraft to observe the actions taken by the system to overcome these faults.
The relation between the parts of the spacecraft and the leon-3 processor can be identified from figure 2.

- Sensors
- Actuators
- Telecommand
- Telemetry

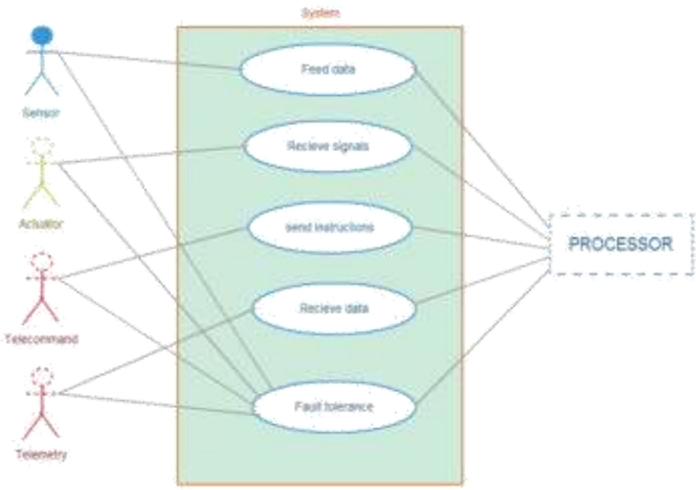


Fig -2: Relation between the parts of spacecraft and processor

4.1 Sensors

A sensor is a transducer whose purpose is to sense (that is, to detect) some characteristic of its environs. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal.

4.2 Actuators

An actuator is a type of motor that is responsible for moving or controlling a mechanism or system.

4.3 Telecommand

A **telecommand** is a command sent to control a remote system or systems not directly connected (e.g. via wires) to the place from which the telecommand is sent.

4.4 Telemetry

Telemetry is the highly automated communications process by which measurements are made and other data collected at remote or inaccessible points and transmitted to receiving equipment for monitoring.

5 ALGORITHM AND METHODOLOGY

A satellite consists of many functional modules which continuously keep interacting with the processor. The main functional modules are :

1. Magnetometer
2. Sun sensor
3. Direction data
4. WDE
5. Wheel speed
6. Temperature sensor
7. Pressure sensor
8. Battery
9. Power sensor
10. RCS elements
11. BDH sync
12. Star sensor
13. Thrusters
14. MTC
15. Heater
16. Reaction wheel
17. IRU

The above specified functions can be identified in Figure 3

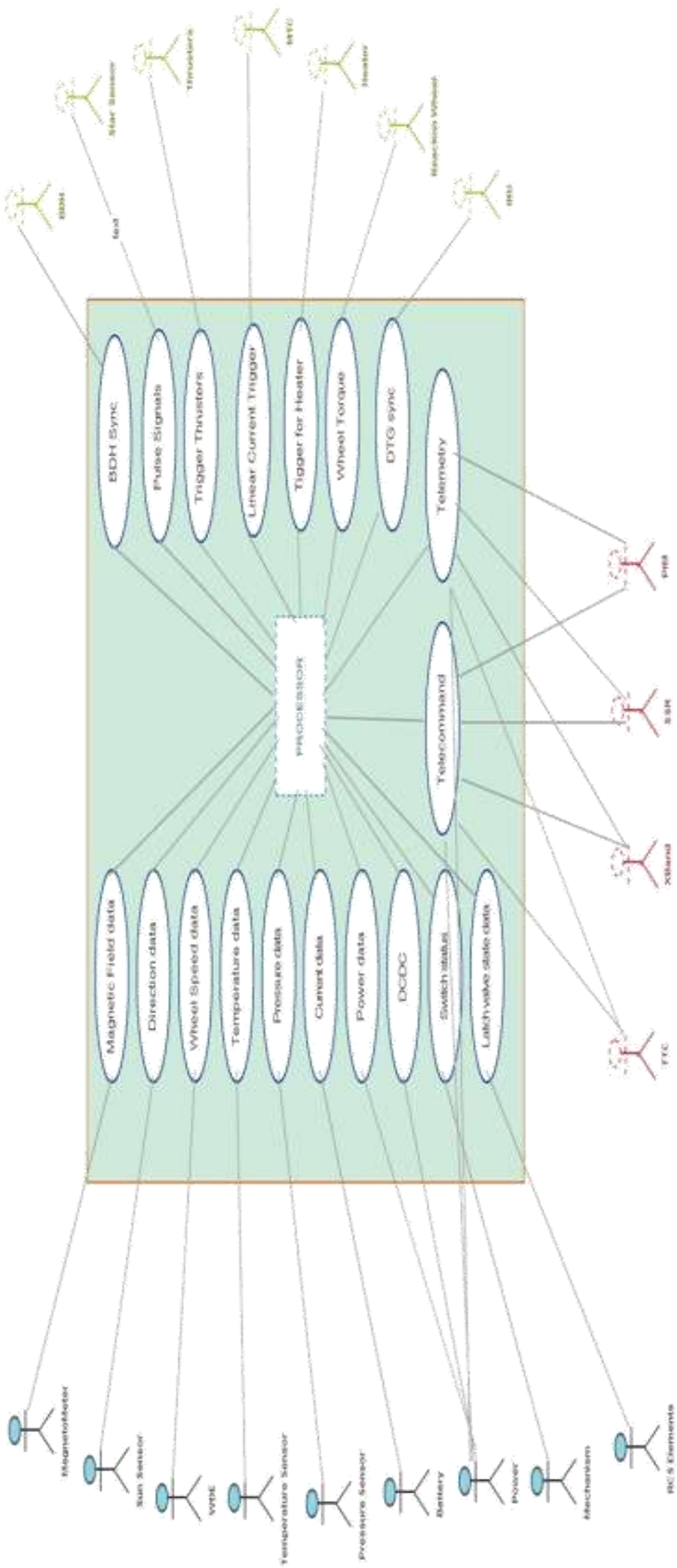
5.1 Working of the LEON-3 Processor

The Leon- 3 processor is the heart the satellite system. It receives data from the sensors and telecommand modules and sends data to the actuators, fault autonomy and telemetry modules.

The processor thus has several tasks in hand and must complete them in a timely manner. Each processor cycle is therefore divided into 4 minor cycles

In the first minor cycle the processor gets data from the sensors. In the second minor cycle, data regarding the health of the satellite is sent to the ground station (telemetry).In the third minor cycle, commands from the ground station are received (telecommand). The final minor clock cycle is left exclusively for fault tolerance.

Fig -3: The main functional modules



International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 02 | May-2015 www.irjet.net p-ISSN: 2395-0072

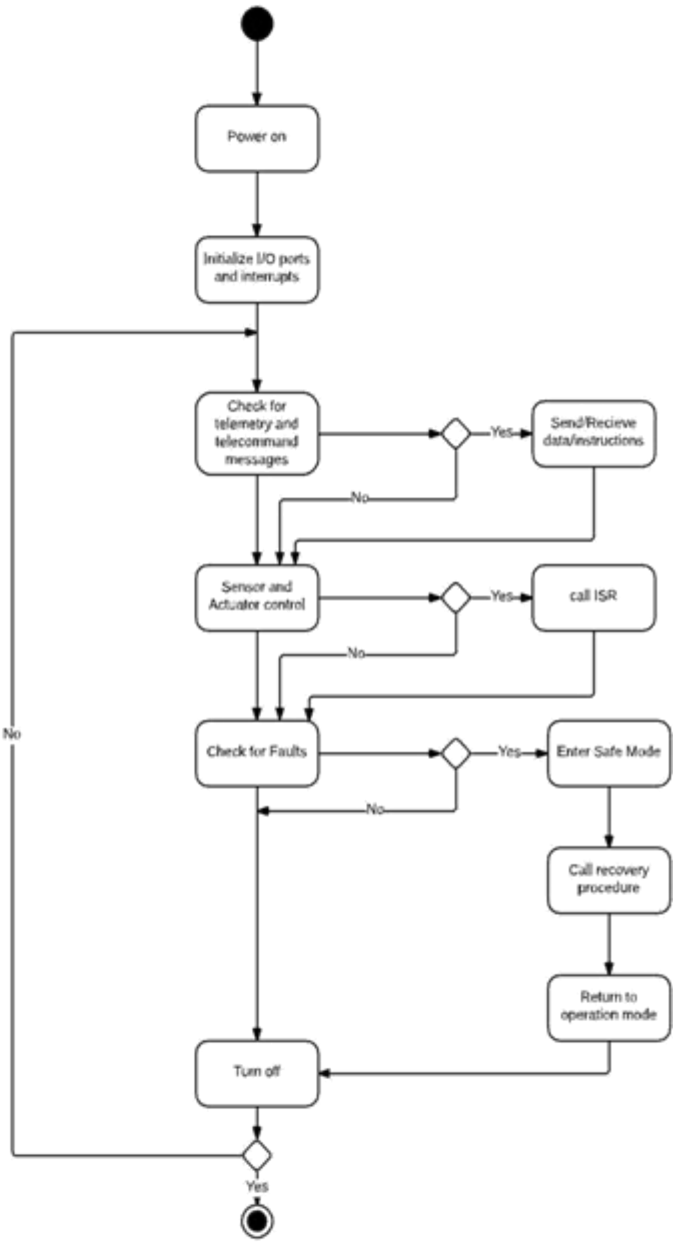


Fig -4: Working of the LEON-3 Processor

5.1 Dealing with Primary Faults

As mentioned previously, the main idea is to identify, locate and treat a fault so that the satellite is brought back to a safe state. The entire satellite system is divided into several modules, as seen in the diagram above. The modules can be categorized mainly into sensors, actuators, LEON-3 processor, telecommand and telemetry.
During the design stages of the satellite, a range of values for parameters such as temperature, pressure and magnetic field etc. are defined. When the satellite is on course a primary fault is said to have occurred if a parameter value is not within the range defined earlier. Thus the sensor module obtains the parameter value, passes it to the LEON-3 processor. The processor in turn calls the fault tolerance module, which checks to see whether the parameter value is out of range. If so, it calls a corresponding actuator, which then brings the satellite back to its safe state.
An example of the above mentioned scenario could be when the temperature sensor of the satellite reads a value above the upper limit of the pre-defined range, in which case a coolant(actuator) will be called, which cools the satellite down. A question arises on how to inject a fault into our perfectly stable Leon-3 simulator? We simply use the debug mode of the tsim and cross compiler and input a value which is out of range for the corresponding parameter.

5.3 Dealing with Secondary Faults

There are a series of events that take place when a parameter value is out of range. What if there is an error in the sensor and not the actual satellite environment ? This would result in a sensor-processor-actuator infinite loop. Thus to prevent these faults (known as secondary faults) we keep redundant sensor systems[4].
These redundant sensors can be divided into two:

5.3.1 Active redundant sensors and passive redundant sensors

Instead of having a single sensor to measure temperature pressure etc, we have three sensors to do the same, known as active redundant sensors. The module that implements fault tolerance, studies the values from each sensor. If a sensor yields large scale irregularities in comparison to the other two sensors, it is rendered faulty. The sensor is then replaced from a set of passive redundant sensors.

6. RESULTS

The simulation of the satellite obtained worked perfectly fine while dealing with primary faults. A problem arises mainly while dealing with secondary sensor faults within satellite system.
It occurs in the following two scenarios:-

1. When more than one active redundant sensors gets damaged.
2. When two faulty active redundant sensors read values that are similar, thus rendering the perfectly healthy sensor faulty.

Also, the TSIM, cross compiler and eclipse combination can successfully be used to simulate any application that requires a LEON-3 processor.

7. FUTURE SCOPE

Our system uses a simple bare board mechanism with a single processor. With further advancements in technology our project could be extended to be used in an RTEMS (Real-Time Executive for Multiprocessor Systems) environment which could results in large scale increase in performance.

8. CONCLUSIONS

Working on an actual LEON-3 processor could work out to be an expensive and complex task. Instead, using TSIM, eclipse and a cross compiler makes it simple and inexpensive.

Satellites usually use SPARC based leon-3 processors for their working. The occurrence of faults in a satellite is inevitable. The method used to rectify the fault may play a vital role in the satellites overall functionality and life time. Having an on- board system that exclusively deals with identification, isolation and reconfiguration of primary and secondary faults would greatly reduce the functions of the ground stations and improve to overall efficiency of the satellite.

REFERENCES

1. http://www.gaisler.com/index.php/products/processors/leon3ft
2. http://www.design-reuse.com/articles/15502/an-implementation-study-on-fault-tolerant-leon-3-processor-system.html
3. http://www.esa.int/Our_Activities/Space_Engineering_Technology/Onboard_Computer_and_Data_Handling/Microprocessors
4. Aviˇzienis, The N-version approach to fault-tolerant soft-ware. IEEE Trans. on Software Eng., SE-11(12):1491–1501, Dec. 1985.

BIOGRAPHIES

Author's Photo

UDIT KHIRAIYA
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

Author's Photo

K.V.S KRISHNA CHAITANYA
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

Author's Photo

NITIN S MALI
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

Author's Photo

VINITKUMAR R.B
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

Author's Photo

JAGADISH KALLIMANI
ASSOCIATE PROFESSOR
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

Author's Photo

PRASHANT KULSHRESTHA
ASSOCIATE PROFESSOR
COMPUTER SCIENCE AND ENGINEERING
MS RAMAIAH INSTITUTE OF TECHNOLOGY

