A Study on HAUSAT-1 Satellite Fault-Tolerant System Architecture Design

Young-Hyun Kim* and Young-Keun Chang**
School of Aerospace & Mechanical Engineering
Hankuk Aviation University, Goyang-City, Korea 412-791

Abstract

A next generation small satellite HAUSAT-1, the first picosatellite developed in Korea, is being developed as one of the international CubeSat program by Space System Research Lab. of Hankuk Aviation University.

A fault-tolerant incremental design methodology has been addressed in this paper. In this study, the effect of system redundancy on reliability was in details analyzed in accordance with the implementation of fault-tolerant system. Four different system recovery levels are proposed for HAUSAT-1 fault-tolerant system optimization. As a result, the HAUSAT-1 fault-tolerant system architecture design and reliability analysis has acquired about 11% reliability improvement.

Key Word : Fault-Tolerant Architecture, Redundancy, HAUSAT-1, Reliability Block Diagram (RBD)

Introduction

CubeSat project was originally adopted in the USSS (University Space System Symposium), and has been promoted mainly by Japan and U.S. universities community.[1] The objective of this project is to provide a standard platform for the design of picosatellites. CubeSats are launched by deployment adapter, Poly-Picosatellite Orbital Deployers (P-POD), designed, fabricated and tested by California Polytechnic State University (Cal Poly). One P-POD is capable of deploying up to three CubeSats. This allows to reduce cost and development time and to enable frequent launches. This project makes colleges and universities from around the world to develop and launch picosatellites without having to interface directly with launch providers.[2]

The CubeSat standard specifies 10cm cubic configuration and 1kg maximum mass. Other additional guidelines are provided by "CUBESAT Design Specifications Document" such as the position of a test port, flight pin, and so on. The specification and design documents are needed to ensure that each satellite will integrate properly with the deployer and neighboring satellites within the deployer and will not interfere with neighboring satellites or, more importantly, the primary payloads or launch vehicle.[3]

The HAUSAT-1 (Hankuk Aviation University SATellite-1) is the first picosatellite developed by university students in Korea. The members of Space System Research Laboratory (SSRL) of Hankuk Aviation University do their best to obtain great achievement, including design and development of the satellite system. One of the mission objective of HAUSAT-1 development is to offer graduate and undergraduate students great opportunities and help them understand the whole development process of satellite design, analysis, manufacturing, assembly, integration, test, launch and operation, and consequently make them specialists in the field of satellite development. Actual mission objectives

* Graduate Student
** Professor
E-mail : ykchang@mail.hau.ac.kr, Tel : 02-300-0286, Fax : 02-3158-3189

in accordance with on-board payload are as followings; collecting the satellite position data with spaceborne GPS receiver, experiment on deployment mechanism of solar cell panel, experiment of homemade sun sensor, and getting data related to satellite Status of Health (SOH) from various sensors.[6]

All satellites are exposed to various harsh space environments. Various kinds of malfunctions and faults can occur frequently at the system and equipment levels. To make matters worse, it is almost impossible to repair and/or maintain the satellite once it is launched. Recent trend of space exploration missions aimed at both high-reliability and low-cost necessitates on-board maintainability or low cost fault-tolerant system. Conventional large satellites having more than 1000kg usually implement the proposed space robust system. On the contrary, small satellites, such as nano and pico level in particular, have difficulties to implement fault-tolerant system since these have limited mass, volume, cost and power budgets.

As mentioned previously, a CubeSat is very small satellite weighing 1kg and having 10 cm cubic size. There are few CubeSats considering fault management system. We, however, tried to incorporate fault-tolerant system discriminately in the HAUSAT-1 design. This paper focuses on the study of fundamental concepts and techniques for designing and analyzing the HAUSAT-1 satellite fault-tolerant system.

HAUSAT-1 Fault-Tolerant System

HAUSAT-1 System Specification

The HAUSAT-1 will orbit at the altitude of perigee 600 km ~ apogee 650(~900) km with 65 or 98 degree inclination angle for one year of design mission life. The HAUSAT-1 is planned to launch in the third quarter of 2004 by the Russian "Dnepr" launch vehicle. The HAUSAT-1 is designed with controller based on standby and warm redundancy. The payloads incorporated are a spaceborne GPS receiver for getting position data, solar panel deployment mechanism to experiment deployment mechanism and generate additional power for GPS, and a homemade sun sensor for space verification. A mass storage memory has 64Mbits volume, which is flash type having nonvolatile characteristic. Average power generated is about 1.5 watt. Communication system has amateur HAM band and UHF/VHF antenna with dipole/monopole type which is made of flexible steel can be folded easily. Table 1 summarizes the HAUSAT-1 system specifications.

The HAUSAT-1 is implementing two types of basic system architectures, both centralized and distributed bus architectures as shown in Fig. 1. The HAUSAT-1 is designed by taking advantages of these two system architectures. The centralized architecture is highly reliable, where failures

Table 1. HAUSAT-1 system specification

Item	Specification	Remark
Altitude	600 ~ 650(900) Km	TBD
Inclination	65 or 98 deg	TBD
Mass	< 1Kg	Payload Included
Size	10 X 10 X 10 cm, Cubic	Payload Included
Power	> 1.5W	@EOL
Attitude Accuracy	5 ~ 7 deg	
Payloads	Spaceborne GPS Receiver Solar Panel Deployment Sun Sensor	
Downlink	2400 bps	FSK
Uplink	1200 bps	FSK
Data Storage	64 Mbits	Flash Type
Mission Life	1 yr	

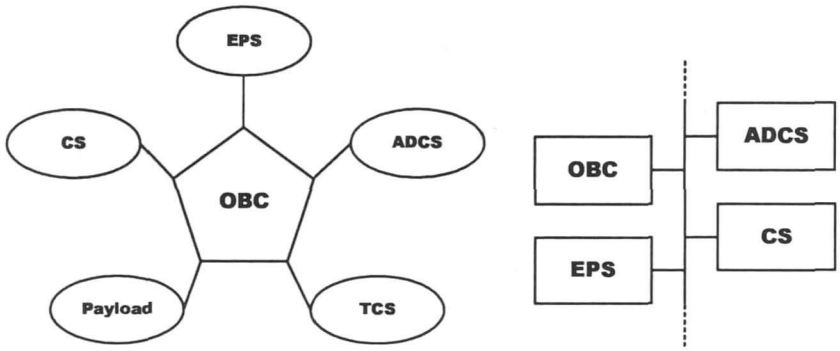


Fig. 1. HAUSAT-1 base architecture

along one interface will not affect the other interfaces.[10] The distributed bus architecture is also highly reliable system because multiple processing units can be used to execute software as needed.[10]

HAUSAT-1 System Architecture

Using the centralized method centering around OBC, the satellite can control, manage and gather various information and data easily. It allows the satellite to rapidly respond to command and various problems unexpected. If a central processing unit gets in trouble, it can give critical influence on the satellite system at one time. It is necessary that back-up or redundancy plan should be considered , specially for the satellite missions. Since distributed bus system architecture provides a high level of redundancy, the weak point of centralized architecture can be overcome by adopting this architecture in parallel. Fig. 2 illustrates HAUSAT-1 hybrid system architecture diagram.

The HAUSAT-1 is designed to implement fault-tolerant system architecture based on ATMEL AVR 8bit controller allowing SPI communication. Fig. 3 represents electrical boards configuration for fault-tolerant system.

The HAUSAT-1 consists of five printed circuit boards; one main board and four sub-boards. One of sub-boards is spaceborne GPSR, manufactured by CMC Electronics in Canada and excluded from fault-tolerant consideration. In Fig. 3, the dotted lines are for monitoring SOH of each subsystem. These lines will be explained later. SPI interface is used for common bus to communicate each other and transmit/receive data.

The HAUSAT-1 will be reconfigured by fault-tolerant procedure to recover the system. The system reconfiguration is organized with OBC as a pivot. The HAUSAT-1 incorporates redundancy to provide fault-tolerant system. Redundancy is a necessary aspect of fault-tolerant system, because

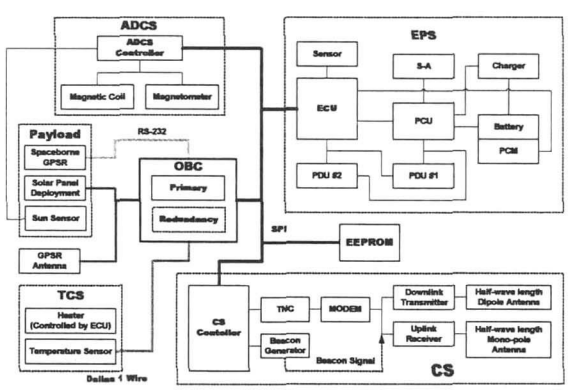


Fig. 2. HAUSAT-1 system architecture diagram

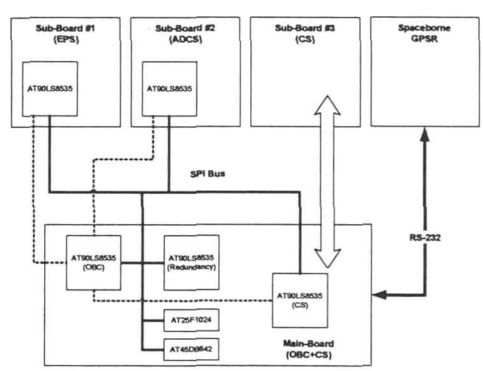


Fig. 3. HAUSAT-1 boards configuration for fault-tolerant system

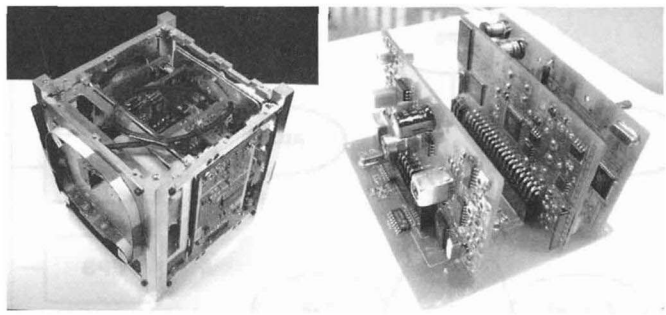


Fig. 4. HAUSAT-1 qualification model

a fault-tolerant system must function correctly even after some of its elements have failed. It is relatively easy to design and incorporate enough redundancy into a system to reduce to acceptably low level of the failure probability which might be due to inadequate element resources. Redundancy circuits and OBC are located in the main board in order to allow easier system reconfiguration.

CS controller can be second redundancy because it is also placed on the main board. The other reason which CS controller is positioned on main board is that CS is very sensitive to electrical and RF noises, so that this subsystem needs well-designed ground and power plane in its sub-board. After all, some digital and controller parts were relocated from the sub-board No. 3 to the main board to assure enough ground and power plane.

Fig. 4 shows HAUSAT-1 system qualification model (left) and its assembled electrical boards (right).

Overview of Fault-Tolerant System

A fault-tolerant system is one that can continue the correct performance of its specified tasks in the presence of hardware and/or software faults. The term fault-tolerant computing is used to describe the process of performing calculations, such as those performed by a computer, in a fault-tolerant manner. Fault-tolerant computing is the art and science of building computing systems that continue to operate satisfactorily in the presence of faults.

The concept of redundancy implies the addition of information, resources, or time beyond what is needed for normal system operation. The redundancy can take one of several forms, including hardware redundancy, software redundancy, information redundancy, and time redundancy.

Fault-Tolerant Algorithm

Fault-Tolerant Plan

Fig. 5 represents HAUSAT-1 fault-tolerant system step. System reset arises depending on watchdog timer and subsystem status. In software redundancy step, flight software is reloaded or replaced with new software. In step 3, system architecture is changed to new architecture design depending on individual case. Step 4 shows system degradation (graceful degradation), which is an important feature, closely related to fault-tolerant system. Graceful degradation is simply the ability of a system to automatically decrease its level of performance to compensate for hardware and software faults.

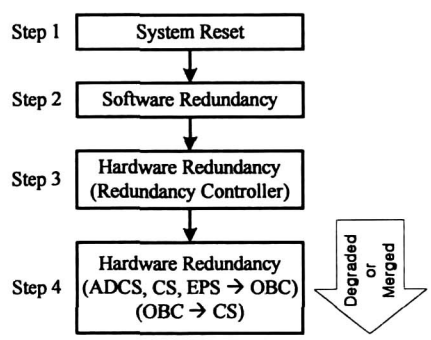


Fig. 5. Fault-tolerant step

System Recovery Levels

Fig. 6 illustrates all fault-tolerant levels and algorithm flows. Each level has the same algorithm flow. Leftside is for watchdog timer reset and rightside is for SOH monitored reset.

Four different system recovery levels are considered for HAUSAT-1 fault-tolerant system. These levels are classified depending on system status and system recovery phase.

Level 1

Level 1 is lowest level recovery performed by watchdog timer reset and monitoring subsystem SOH to prepare for reset enforcement and software reloading. In level 1, the satellite system is almost in normal operation. No system change is required at this level.

Level 2

This recovery level is similar to level 1. But there is one big difference from level 1. Whereas the program reloading occurs in level 1, the flight software is replaced from primary software to redundant software in level 2. The system architecture is also changed to be able to use redundancy controller. The HAUSAT-1 possesses three different flight softwares such as primary, redundant and degradation flight softwares.

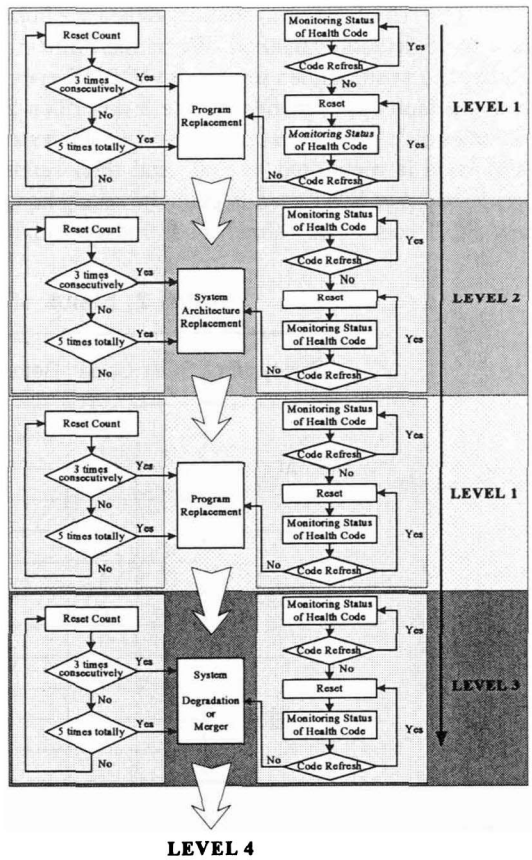


Fig. 6. Fault-tolerant algorithm flow

Level 3

In level 3 recovery, the satellite system is degraded or merged to simple system. In this level, flight software is changed into degradation software.

Level 4

This is the last of HAUSAT-1 fault-tolerant levels. In level 4, there are only limited ways to recover system from fault or error. The watchdog timer reset and SOH of subsystem monitor can be performed. Once the satellite goes into this level, the system is kept without any other processing for recovery until failure.

System Recovery Method

The HAUSAT-1 is implementing system monitor for fault detection, and watchdog timer reset, software reloading/replacement, and system architecture change for system reconfiguration and system recovery to initial system. The HAUSAT-1 recovery is based on redundancy. Redundancy means the existence of more than one method to perform a required function.[8] Redundancy does not just imply a duplication of hardware because it can also often be implemented by coding, at the software level. or in another form.[8]

System Monitor

The HAUSAT-1 system monitors the number of reset by watchdog timer and subsystem SOH as a fault-detecting method. When the number of reset is three times consecutively or five times totally, the system goes to next level. SOH code of each subsystem should be changed or refreshed to avoid fault level change. Table 2 describes SOH codes in the subsystem. Refresh code is used to distinguish from SOH code when the subsystem sends the same SOH code. For example, if ADCS SOH code is monitored by "00" and then sends again the same SOH code next time, OBC must decide whether SOH code is changed or unchanged. The refresh code is changed whenever sending new SOH code. The refresh code makes judgment easy in this fashion.

Table 2. Status of health code of subsystem

Subsystem	SOH Code		Refresh Code	Description
ADCS	0	0	0/1	Good
	0	1	0/1	Need Attitude Control
	1	0	0/1	
	1	1	0/1	
CS	0	0	0/1	Beacon Mode
	0	1	0/1	Downlink Mode
	1	0	0/1	Uplink Mode
	1	1	0/1	
EPS	0	0	0/1	Good
	0	1	0/1	Battery Warning Level 1
	1	0	0/1	Battery Warning Level 2
	1	1	0/1	Emergency Operation

Watchdog Timer Reset

The HAUSAT-1 basic fault-tolerant method is reset. The reset is occurred by watchdog timer. Watchdog timer is a kind of incremental counter and must be reset by controller periodically. If the controller fails to reset watchdog timer caused by permanent delay or infinite loop, an overflow occurs and then triggers a local reset.

Software Reloading/Replacement

When the reset occurs 3 times consecutively or 5 times totally, the flight software of satellite system will be reloaded. This process will also be performed when the subsystem SOH is not updated to new data.

Software replacement is leading to system architecture change. Since the system is still unstable even after the software is reloaded, the satellite needs to replace new flight software to be able to implement new hardware architecture.

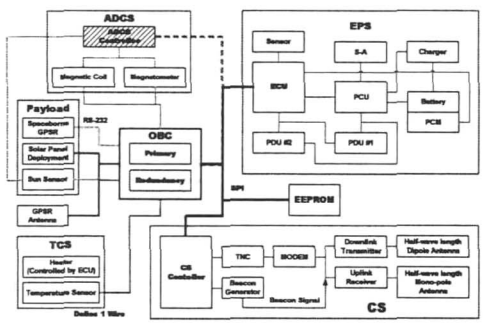
The HAUSAT-1 has three kinds of software. The primary flight software is for normal operation. The software used in level 1 to reload is primary flight software. In level 2, the system is reconfigured and followed by requiring new software. The redundant software is needed for new hardware system architecture with redundancy controller. The HAUSAT-1 has one more reconfigured system which graceful degradation is considered.

System Architecture Change

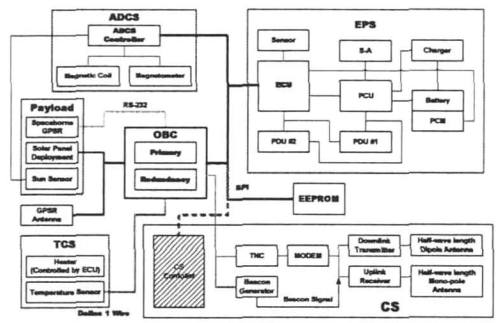
If the system has still faults even after the software is reloaded, the satellite shall consider this situation as hardware fault. The satellite system will be changed to new architecture according

to designated plan associated with each subsystem. If the system requires architecture change, the satellite will try to use redundancy at first and then will be converted to degradation system.

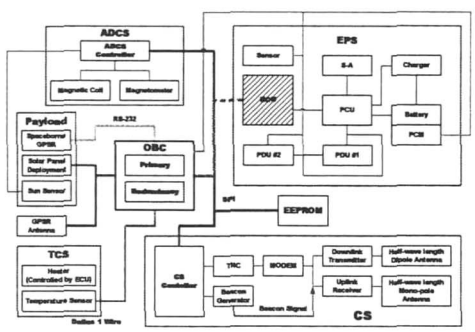
Fig. 7 presents designated recovery system architecture for individual subsystem. Fig. 7(a) is for ADCS recovery plan, and Fig. 7(b) is designed to protect CS failure. Fig 7(c) is EPS recovery system architecture. Figures 7(d) and 7(e) are prepared for OBC, which has two fault recovery architectures to provide more reliable system. ADCS, CS and EPS are recovered by OBC, and OBC is restored by redundancy and CS.



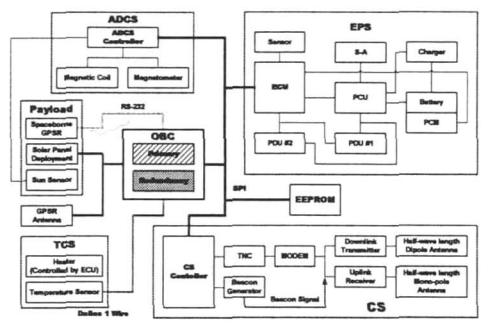
(a) ADCS



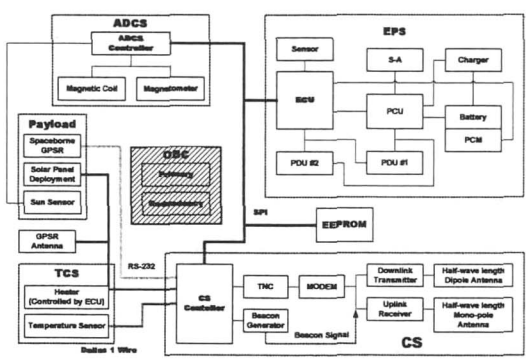
(b) CS



(c) EPS



(d) OBC (Case I)



(e) OBC (Case II)

Fig. 7. Recovery system architecture

System Restore

In level 1, 2 and 3, if the satellite system operates well without any problems the satellite will try to restore into initial system. This process is not performed automatically, since this process is very risky. In this case, system restore is carried out by command from ground station.

HAUSAT-1 System Reliability Analysis and Results

Reliability is a characteristic of an item, expressed by the probability that the item will perform its required function under given conditions for a stated time interval. It is generally designated by R. From a qualitative point of view, reliability can be defined as the ability of an item to remain functional. Reliability specifies thus the probability that no operational interruptions will occur during a stated time interval. This does not mean that redundant parts may not fail, such parts can be failed and repaired. The concept of reliability thus applies to nonrepairable as well as repairable items. To make sense, a numerical statement of reliability must always be accompanied by the definition of the required function, the operating conditions, and the mission duration.

Many engineers are designing large, complex, and sophisticated systems and require knowledge of many specific areas of reliability. Furthermore, the operation engineers have to maintain such systems in the field with the minimum maintenance costs and maximum availability. Therefore, the knowledge of basic reliability and various specific areas of reliability are essential to these engineers and other personnel closely related to systems design and their operation in the field. By keeping this in mind, this study is an attempt to fulfill this specific need because there has been a considerable growth of knowledge in several specific areas of reliability and its applications.[9]

This reliability analysis may be the first attempt in CubeSats programs. Generally, the system reliability analysis has not been performed for small satellites utilizing COTS (Commercial Off-The-Shelf) parts because of the difficulty of getting generalized failure rate value. This study has been initiated to analyze how the reliability is increased in new system by adopting fault-tolerant compared to non fault-tolerant system. In this section, the effect of system redundancy on reliability was in detail analyzed in accordance with the implementation of fault-tolerant system.

Reliability Analysis Plan

Reliability Block Diagram (RBD)

The Reliability Block Diagram (RBD) is an event diagram. Setting up an RBD involves first partitioning the item into elements with clearly defined tasks. The elements which are necessary for the required function are then connected in series, while elements which can fail with no effect on the required function (redundancy) are connected in parallel [Refer to Fig. 8]. Obviously, the ordering of the series elements is arbitrary.[8]

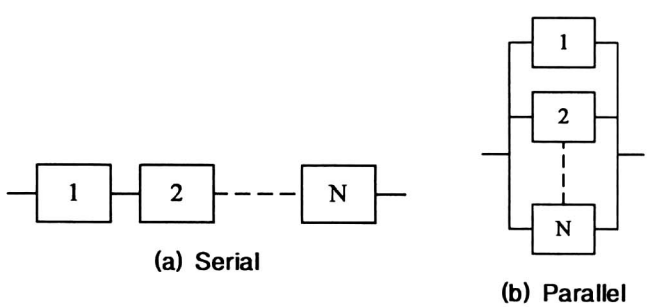


Fig. 8. Reliability block diagram

Basic Reliability Functions

Equation (1) is reliability function of one-item structures with constant failure rate assumed. λ is constant failure rate and t is operating time.

$$R(t) = e^{-\lambda t} \tag{1}$$

Reliability of Serial Structures

Reliability of serial structures can be expressed by following equations.

$$R_{SS}(t) = R_1(t) \cdot R_2(t) \cdots R_N(t) = \prod_{i=1}^{N} R_i(t) \tag{2}$$

$$R_{SS}(t) = e^{-\lambda_1 t} \cdot e^{-\lambda_2 t} \cdots e^{-\lambda_N t} = e^{-\left(\sum_{i=1}^{N} \lambda_i\right) t} \tag{3}$$

$$\lambda_S = \sum_{i=1}^{N} \lambda_i \tag{4}$$

In series structures, the failure rate of an item (equipment or system) without redundancy that consists of independent element is equal to the sum of the failure rates of its elements as expressed by equation (4).[8]

Reliability of Parallel Structures

Reliability of parallel structures can be expressed by following equations.

$$R_{SP}(t) = 1 - (1 - R_1(t))(1 - R_2(t)) \cdots (1 - R_N(t)) \tag{5}$$

$$= 1 - \prod_{i=1}^{N} (1 - R_i(t)) \tag{6}$$

$$= 1 - \prod_{i=1}^{N} (1 - e^{-\lambda t}) \tag{7}$$

Parallel structures can be used to improve system reliability. This system will only fail if all of its components fail.

Assumptions for Reliability Calculation

Reliability analysis is a complicated work. Since HAUSAT-1 uses COTS parts, it is difficult to know exact failure rate. The satellite system reliability relies on EEE parts, redundancy implemented, etc. The following assumptions are made for this reliability analysis. These assumptions allow the reliability calculation to become easy and simple.

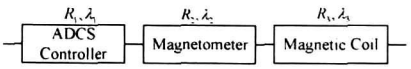
① Constant failure rate ($\lambda(t) = \lambda$)
② All parts and modules have the same failure rate and the same reliability as 0.9 (e.g. R=0.9)
③ In parallel structures case, active redundancy is applied.
④ Major parts such as controller, memory, sensor, etc. are considered to calculate reliability. (EEE parts such as resistor and diode are neglected.)
⑤ All parts are COTS.
⑥ Thermal and environmental characteristics are not considered. (Only system architecture is considered for reliability calculation)

Effect of Redundancy on Reliability

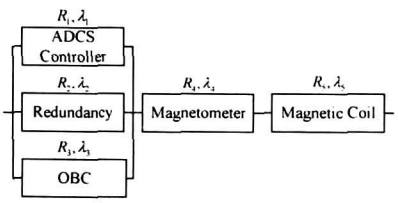
In this chapter, it shows reliability calculation of individual subsystem by using reliability block

diagram and then compares between basic system and redundant system.

Attitude Determination & Control Subsystem (ADCS)



(a) RBD w/o redundancy



(b) RBD w/ redundancy

Fig. 9. ADCS reliability block diagram

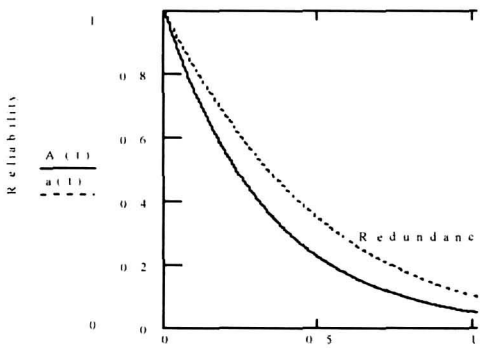


Fig. 10. ADCS comparison between with and without redundancy

In Fig. 9(a)

$$R = R_1 = R_2 = R_3 = 0.9$$

$$R = R_1 \cdot R_2 \cdot R_3$$

$$= 0.729$$

$$\lambda = \lambda_1 = \lambda_2 = \lambda_3$$

$$A(t) = e^{-3\lambda t}$$

In Fig. 9(b)

$$R = R_1 = R_2 = R_3 = R_4 = R_5 = 0.9$$

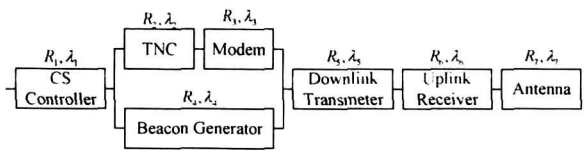
$$R = \left[1 - (1 - R_1)(1 - R_2)(1 - R_3)\right] \cdot R_4 \cdot R_5$$

$$= 0.809$$

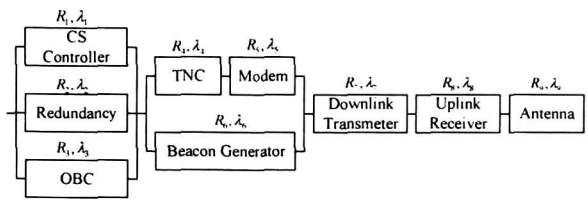
$$\lambda = \lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = \lambda_5$$

$$a(t) = e^{-2\lambda t} \cdot \left[1 - \left(1 - e^{-\lambda t}\right)^3\right]$$

Communication Subsystem (CS)



(a) RBD w/o redundancy



(b) RBD w/ redundancy

Fig. 11. CS reliability block diagram

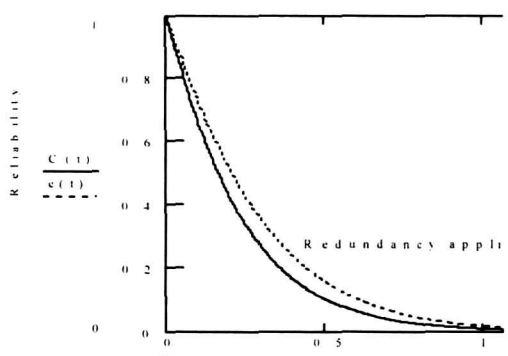


Fig. 12. CS comparison between with and without redundancy

In Fig. 11(a)

$$R = R_1 \cdot [1 - (1 - R_2 \cdot R_3)(1 - R_4)] \cdot R_5 \cdot R_6 \cdot R_7$$
$$= 0.644$$

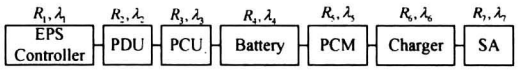
$$C(t) = e^{-4\lambda t} \cdot [1 - (1 - e^{-2\lambda t})(1 - e^{-\lambda t})]$$

In Fig. 11(b)

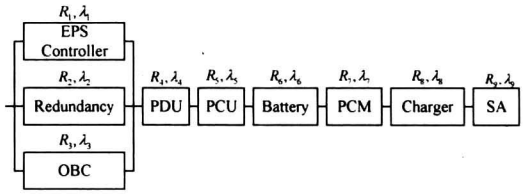
$$R = [1 - (1 - R_1)(1 - R_2)(1 - R_3)] \cdot [1 - (1 - R_4 \cdot R_5)(1 - R_6)] \cdot R_7 \cdot R_8 \cdot R_9$$
$$= 0.714$$

$$c(t) = e^{-3\lambda t} \cdot [1 - (1 - e^{-\lambda t})^3][1 - (1 - e^{-2\lambda t})(1 - e^{-\lambda t})$$

Electrical Power Subsystem (EPS)



(a) RBD w/o redundancy



(b) RBD w/ redundancy

Fig. 13. EPS reliability block diagram

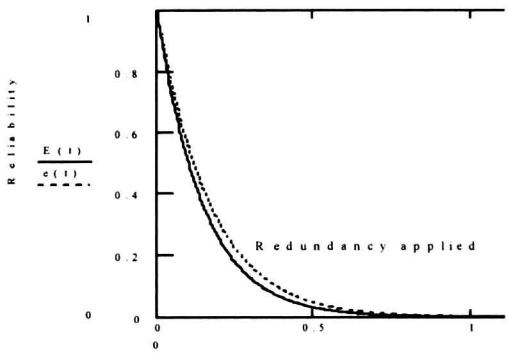


Fig. 14. EPS comparison between with and without redundancy

In Fig. 13(a)

$$R = R_1 \cdot R_2 \cdot R_3 \cdot R_4 \cdot R_5 \cdot R_6 \cdot R_7$$
$$= 0.478$$

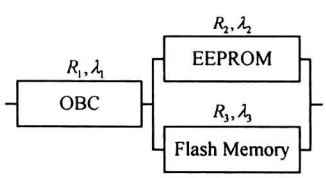
$$E(t) = e^{-7\lambda t}$$

In Fig. 13(b)

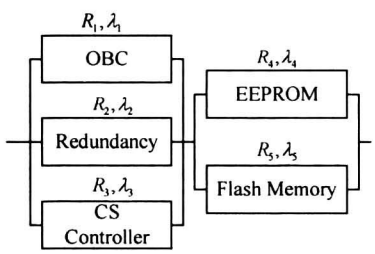
$$R = [1 - (1 - R_1)(1 - R_2)(1 - R_3)] \cdot R_4 \cdot R_5 \cdot R_6 \cdot R_7 \cdot R_8 \cdot R_9$$
$$= 0.531$$

$$e(t) = e^{-6\lambda t} \cdot [1 - (1 - e^{-\lambda t})^3]$$

On-Board Computer Subsystem (OBC)



(a) RBD w/o redundancy



(b) RBD w/ redundancy

Fig. 15. OBC reliability block diagram

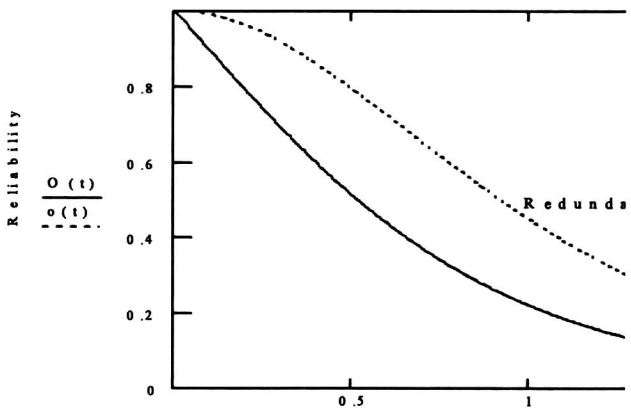


Fig. 16. OBC comparison between with and without redundancy

In Fig. 15(a)

$$R = R_1 \cdot [1-(1-R_2)(1-R_3)]$$
$$= 0.891$$
$$O(t) = e^{-\lambda t}\left[1-\left(1-e^{-\lambda t}\right)^2\right]$$

In Fig. 15(b)

$$R = [1-(1-R_1)(1-R_2)(1-R_3)]\cdot[1-(1-R_4)(1-R_5)]$$
$$= 0.989$$
$$o(t) = \left[1-\left(1-e^{-\lambda t}\right)^3\right]\left[1-\left(1-e^{-\lambda t}\right)^2\right]$$

The RBDs with redundancy are compared to RBDs without redundancy for individual subsystem in Figures 9, 11, 13, and 15. All subsystem rates of reliability growth are about 11% when the redundancy is integrated in their system. The reliability analysis results are presented in Figures 10, 12, 14 and 16 for individual subsystem, respectively. As shown in Fig. 16, the reliability in OBC subsystem is relatively, gradually decreased as λt increases in comparison to other subsystems. It means that the OBC is more parallelized than others from the viewpoint of system architecture. Fig. 17 represents the relative comparison of reliability among subsystems with redundancy.

The reliabilities of all subsystems except for OBC are decreased compared to individual module reliability of 0.9. The reliability of EPS is the lowest among subsystems. Comparing redundant EPS with 0.9 which is assumed to be baseline reliability, the reliability reduction of EPS is 41% as shown in Table 3. In the case of EPS there are many modules connected serially whose reliabilities have small values. Through this analysis, it was found that CS and EPS are frail and need to consider reliability enhancement.

Table 3. Reliability analysis results

Subsystem	Non-Redundancy	Redundancy	Reliability Growth	Redundancy/0.9
ADCS	0.729	0.809	10.97%	89.89%
CS	0.644	0.714	10.88%	79.33%
EPS	0.478	0.531	11.09%	59%
OBC	0.891	0.989	11%	109.89%

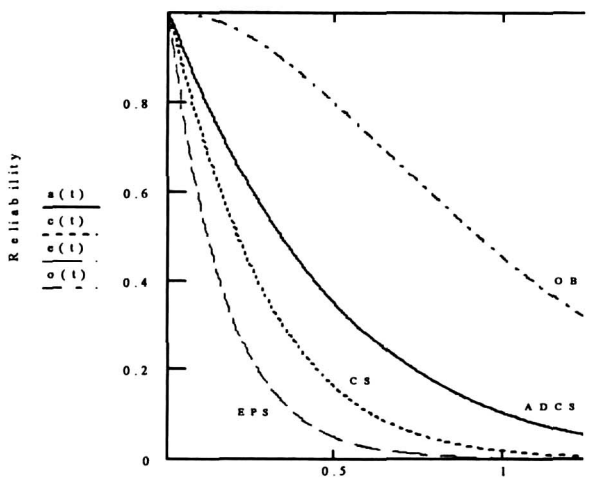


Fig. 17. Comparison among subsystems with redundancy

The reason which redundant electrical module is not considered for HAUSAT-1 fault-tolerant system is to make the system as simple as possible. A complicated system is one of the factors which make reliability reduce. These results are very helpful to investigate which subsystem is fragile and more trade-off might be required to make better reliable system.

Conclusions

Avoiding faults and errors in satellite computing system is an important issue since maintenance is not almost feasible in space. Satellites have typically long-life application. They have usually reliability requirements with a high probability of being operational at the end of life. The main goal is to keep the computers operational during mission life time.

Fault-tolerance is achieved by applying a set of analysis and design techniques to create systems with dramatically improved dependability. As new technologies are developed and new applications arise, new fault-tolerance approaches are also needed. Many applications and systems related to fault-tolerance were introduced for large satellite system. But there are few applications and system for picosatellite.

A fault-tolerant incremental design methodology has been presented in this paper. In this study, four redundant levels optimized the HAUSAT-1 are proposed.

The HAUSAT-1 fault-tolerant system architecture design and reliability analysis demonstrate that it is possible to increase the reliability of a COTS based picosatellite system operating in the space environment. It is found that a reliability consideration becomes very important in the planning, design and operation of satellite system in space particularly. It is achieved that the HAUSAT-1 fault-tolerant system has acquired about 11% reliability improvement. The new system brought forward in this paper is based on redundancy and sharing controllers in case of graceful degradation.

Acknowledgement

This study is supported by Core Space Technology Program and partially by National Research Lab. (NRL) Program funded by Ministry of Science and Technology. The authors would like to appreciate for the financial support.

References

1. http://dat15.t.u-tokyo.ac.jp/cubesat/about/index-e.html.

2. Jake A. Schaffner, "The Electronic System Design, Analysis, Integration, and Construction of the Cal Poly State University CP1 CubeSat," 16th AIAA/USU Conference on Small Satellites.

3. "CUBESAT Design Specifications Document," Revision VII, February, 2003.

4. Mr. Hank Heidt, Prof. Jordi Puig-Suari, Prof. Augustus S. Moore, Prof. Shinichi Nakasuka, Prof. Robert J. Twiggs, "CubeSat: A new Generation of Picosatellite for Education and Industry Low-Cost Space Experimentation," 14th Annual/USU Conference on Small Satellites, 2000.

5. Young-Hyun Kim, Seung-Won Seo, Nam-Sook Chung and Young-Keun Chang, "Three-Axis Attitude Stabilization System Design of Picosatellite HAUSAT-1," CubeSat Symposium Papers, 1st International CubeSat Symposium, Tokyo, Japan, 2003.

6. Young-Hyun Kim and Young-Keun Chang, "Development of HAUSAT-1 Picosatellite (CubeSat) for Educational Purpose," CubeSat Symposium Papers, 1st International CubeSat Symposium, Tokyo, Japan, 2003.

7. Dhiraj K. Pradhan, "Fault-tolerant computer system design," 1995, Prentice Hall.

8. A. Birolini, "Reliability Engineering Theory and Practice," 3rd Ed. Springer.

9. Baldir S. Dhillon, "Reliability Engineering in Systems Design and Operation," 1983, Van Nostrand Reinhold Company Inc.

10. James R. Wertz and Wiley J. Larson, "Space Mission Analysis and Design," 3rd Ed. Space Technology Library.

11. D. C. Ionescu and N. Limnios, "Statistical and Probabilistic Models in Reliability," Birkhäuser.

12. Dimitri Kececioglu, "Reliability Engineering Handbook," Volume 1, 1991, Prentice-Hall.

13. John W. Evans and Jillian Y. Evans, "Product Integrity and Reliability in Design," Springer.

14. Wallace R. Blischke and D. N. Prabhakar Murthy, "Reliability Modeling, Prediction, and Optimization," Wiley Inter-Science.