A Full-Digital Test Method for Satellite-Borne Software

Pan Xin[1](✉), Jiao Rong-Hui[2], and He Jing[2]

[1] Institute of Spacecraft System Engineering, Beijing 100094, China
228192692@qq.com, 4028212@qq.com, 549105167@qq.com
[2] China Academy of Space Technology, Beijing 100094, China

Abstract. In order to realize satellite complex and flexible function, reduce the satellite development cycle and cost, and realize modular production, software in satellite occupies a pivotal role. Due to the powerful software function, flexible configuration and convenient update, the software test is more and more complex. The current star borne software test is mainly based on the real hardware environment, with software and hardware serial development, which consumes more resources. This paper presents a soft condition based on a full-digital test environment. The test method, achieving software test and hardware decoupling, realizes the parallel development of software and hardware, and shortens the development cycle. The test efficiency increases by more than 25%.

Keywords: Satellite-borne software · Testing · Digital

1 Foreword

With the advent of the fourth industrial revolution, modern industry is developing towards the direction of intelligent network, and software plays a decisive role in the current industrial production process.

The communication equipment software test of domestic Huawei companies is mainly based on the computer hardware simulation framework GEM5, the self-developed hardware simulation environment test system AutoSpace, and the communication equipment software is tested based on the hardware simulation environment. After the completion of the test, the injection hardware directly carries out EMC and other tests. Alibaba's test team recently used the artificial intelligence automated test tool EggPlant to carry out software testing, EggPlant is an automated testing tool for HMI (Human-computer Interaction system), which is based on artificial intelligence. Eggplant is based on interface modeling, locks the page status, and automatically generates test cases. The algorithm optimizes potential, massive test case series to prioritize high-value test cases. In model-driven testing under AI-based optimization, the test path for each test is determined based on the data results of the previous test [1].

The software testing of foreign SpaceX companies has a high level of automation, and can use DevOps high automation development mode in spacecraft testing, which The level of automated testing benefits from virtual testing environment technology and advanced automated testing framework. The software debugging and testing tools of ARM company are based on SIMICS hardware simulation platform, maintaining the development process of "hardware chip design and software simulation development in parallel, and hardware product and software product development in parallel" [2] (Fig. 1).

code-for-a-living MAY 11, 2021

Testing software so it's reliable enough for space

We've talked about the engineers who write the code that operates SpaceX spaceships. Now let's talk about the people who build and maintain the tools and processes that enable the developers and ultimately, help accomplish the mission of flying astronauts to space. Stack Overflow talked with Erin Ishimoticha, an engineer in the Software Delivery Engineering...

Fig. 1. Software testing by the SpaceX Company.

From the research situation, the virtual software based on full digital test environment has great advantages. First of all, it can improve the test coverage, enhance the business logic test coverage, and improve the ability of fault condition verification and problem recovery and investigation. Secondly, it can shorten the development cycle, decouple software testing and hardware development, and carry out software and software development in parallel. Finally, it can reduce the test cost, establish a flexible testing environment, leave the hardware testing equipment, and improve the test standardization.

2 Full-Digital Software Testing Environment Construction

The full digital software test environment mainly includes the digital test platform, the software test external input model and the test data processing. Test data processing, similar to hardware-based testing, is not introduced in this paper. The full-digital software test environment framework is shown in Fig. 2.

2.1 Digital Test Platform Construction

The digital test platform is built based on the runtime core (Kernel), which controls the operation of the whole platform, including scheduling the operation of various simulation models, and controlling discrete events such as clock or interruption. The standard interface is defined in the platform. Various simulation models and various simulation models accept the scheduling of the runtime core through this interface to complete the data exchange between the models. The simulation model constitutes a fully digital software test environment under the unified scheduling of the runtime core. The tested target software can be run in the fully digital software test environment without any modification [3].

The simulation model in the digital test platform is available Package it into an independent Windows dynamic link library to quickly build a virtual test system by "building" it. The digital test platform has established the common CPU model library of the embedded processor and the common chip simulation model library. When building a digital test environment, you only need to select the required model library from the existing simulation model library and make the necessary configuration to avoid repeated development [4].

In the human-machine interface layer, the digital test platform integrated environment interface is configured to complete software debugging, interface display and other tasks. The target code analysis library is also configured to analyze the measured soft. The compiled target file and test coverage analysis tool are used to analyze the software test coverage.

At the same time, the test platform provides extended functions. Users can develop their own simulation models and gradually establish commonly used simulation model libraries, so as to reduce the development workload of building a digital test environment and improve the test efficiency [5, 6].

2.2 External Input Modeling

The main function of the software test external input model is to provide the external incentive of the software for the tested load software, and respond to the execution effect of the tested software control instructions, so as to realize the upstream and downstream connection of the data. The external input modeling of satellite-load software mainly includes satellite dynamics simulation model, satellite bus terminal simulation model, user terminal simulation model, etc.

Satellite dynamics simulation model includes orbit and attitude simulation model, sensor simulation model and actuator simulation model, etc. The input parameters of orbit and attitude simulation model are mainly orbit, attitude configuration parameters, time T, thrust, acceptance moment, whole star mass characteristic parameters, etc., and the output parameter is satellite orbit and posture in time T. The sensor simulation model is the position relationship of external pose and sensitive target, and the output parameter is azimuth. The actuator simulation model input is the execution pulse, the output is the execution torque, etc.

Satellite bus terminal simulation model mainly simulates various kinds of terminals attached to the satellite bus, including 1553B bus terminal model, CAN bus terminal model, CSB bus terminal model and RS422 serial bus terminal model. Data interaction with the tested software according to the bus communication protocol and terminal functions. Input to the user needs, technical requirements, etc., and output to reflect the functions and Telemetry parameters of the technical indicators.

The user terminal simulation model mainly simulates the ground terminal to complete the large loop function and performance test [4].

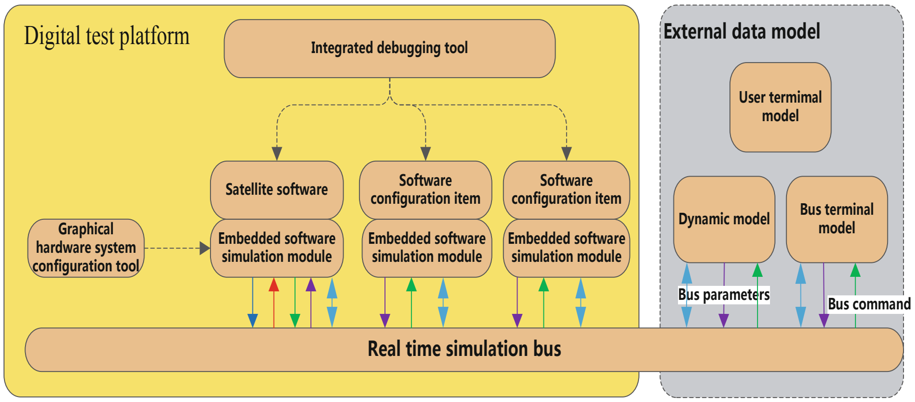
Digital test platform
Integrated debugging tool
Satellite software
Software configuration item
Software configuration item
Graphical hardware system configuration tool
Embedded software simulation module
Embedded software simulation module
Embedded software simulation module
Real time simulation bus
External data model
User termimal model
Dynamic model
Bus terminal model
Bus parameters
Bus command

Fig. 2. Composition diagram of the star-loaded full-digital software test environment.

3 Test Case Design

Test case design should improve the efficiency and quality of satellite software, adopt more advanced automatic test framework, improve the logic test capability, supplement the whole satellite fault test without hardware damage and orbit fault logic verification; supplement the satellite system level test, test the shunt software system, and improve the system efficiency. Based on the advantages of this system, the branch coverage test of the software system is emphasized to provide a supplement for the satellite software test [7, 8].

4 Situation of Application

4.1 Posture and Orbit Control Software Test

The geosynchronous orbit (GEO) satellite posture and orbit control software is selected as the test object, and the full digital software test environment is built based on the digital test platform and dynamic simulation model. The IPM inertial pointing mode test was performed, and the test results are shown in Figs. 3, 4, 5, and 6. As can be seen from the figure, in IPM mode, the control sensitivity uses star min + top, and the actuator selects the thruster. The attitude angle convergence can be stabilized in the control accuracy, which is basically consistent with the dynamic simulation data.

4.2 Energy Management Software Testing

The geosynchronous orbit (GEO) satellite energy management software is selected as the test object, and the full digital software test environment is built based on the digital test platform and the bus terminal simulation model. The battery pack charge and discharge logic test and the energy FDIR processing strategy test were conducted. The test results meet the requirements, as shown in Tables 1 and 2.As can be seen from the table, the test results meet the requirements, the test efficiency battery charge and discharge logic test increased by about 30%, and the energy FDIR processing strategy test increased by about 25%.

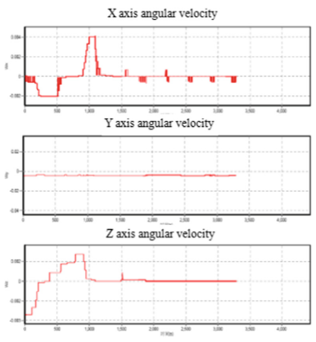

Fig. 3. Upper angular velocity of a satellite star.

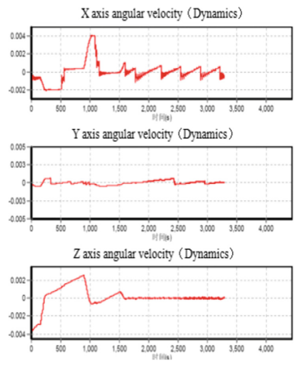

Fig. 4. Dynamic simulation of the angular speed.

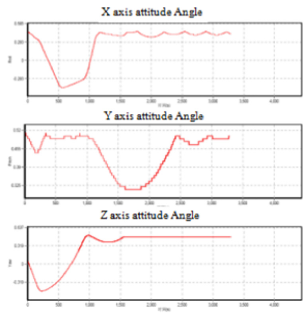

Fig. 5. Position Angle of a satellite star.

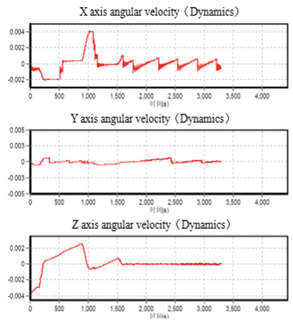

Fig. 6. Dynamics simulation attitude angle.

Table 1. Comparison table of battery pack charge and discharge logic test results of GEO Satellite Energy Management Software.

Name of test	Test requirements	Satellite hardware environment (min)	Digital Environment (min)	Test conclusion
Initialize the telemetry parameter check	√★▲	20	10	Correct
Software telemetry parameter modification function check	√★▲	120	80	Correct
Automatic statistical calculation of the PCU telemetry parameters	√★▲	90	60	Correct
Automatic statistical calculation of the BIMU telemetry parameters	√★▲	60	30	Correct

(*continued*)

Table 1. (*continued*)

Name of test	Test requirements	Satellite hardware environment (min)	Digital Environment (min)	Test conclusion
Battery overcharging autonomous setting check	√ ★ ▲	20	10	Correct
Sun /Earth (Moon) shadow sign switch and battery independent charging setting	√ ★ ▲	150	110	Correct
Real-time calculation of battery charge state and discharge depth	√ ★ ▲	40	30	Correct
TD accumulation of a single discharge time and telemetry downward transmission	√ ★ ▲	120	90	Correct
Balance processing logic check of the battery pack (Main configuration of light period)	√ ★ ▲	120	90	Correct
Balance processing logic check of the battery pack (Main configuration of Earth shadow period)	√ ★ ▲	120	90	Correct

Table 2. Comparison of Energy FDIR Strategy Test Results of GEO Satellite Energy Management Software.

Name of test	Test requirements	Satellite hardware environment (min)	Digital Environment (min)	Test conclusion
Battery pack charging voltage setting instruction correctness judgment function check	√ ★ ▲	100	80	Correct
Function of battery charging current setting instruction	√ ★ ▲	100	80	Correct
Check the charging function of battery pack	√ ★ ▲	90	60	Correct

(*continued*)

Table 2. (*continued*)

Name of test	Test requirements	Satellite hardware environment (min)	Digital Environment (min)	Test conclusion
Battery pack high and low voltage monitoring and alarm	√★▲	180	140	Correct
S3R module protection and reset real-time monitoring function check	√★▲	120	90	Correct
BCDR module autonomous protection implementation of monitoring function inspection	√★▲	120	90	Correct
Bus overvoltage protection load accidentally turn on the real-time monitoring function inspection	√★▲	30	20	Correct
Battery DOD1Fault handling function check	√★▲	30	20	Correct

√ Test project required by the test outline, ★ based on satellite hardware test project, ▲ based on digital software test environment test project.

5 End

This paper proposes a satellite software test method based on full digital test environment, which realizes the parallel development of satellite software and satellite hardware products, and shortens the satellite development cycle. In addition, the software test based on the full digital test environment can be carried out on the ordinary PC, which greatly saves resources, shortens the instruction execution and telemetry return time, and improves the test efficiency. In the future, the simulation time of the full digital test environment can be further improved, the accelerated test of satellite software can be realized, and the test efficiency will be further improved.

References

1. Urumqi, et al.: Virtual test method and application for spacecraft system-level testing, J. Avia. **7** (2017)
2. Hu, K., et al.: Virtual test technology research. In: 2016 National Metrology and Testing Technology Academic Exchange Conference, vol. 5 (2016)

3. Zhu, Y., et al.: Satellite control system embedded software virtualization test platform technology research. Master of Engineering Dissertation, Shanghai Jiao Tong University, vol. 10 (2014)
4. Songqing, L.: Research on General Simulation Technology of Spacecraft Control System, p. 2. National Defense University of Science and Technology, Changsha (2012)
5. Cui, W., Yang, H., Yang, J., Liu, J.: Design and implementation of satellite subsystem simulation test platform. Comput. Meas. Control 10 (2015)
6. Qin. J., et al.: Satellite computer software testing platform, Master thesis, School of Software, Tongji University (2007). 6
7. Huamao, W.: Spacecraft Comprehensive Test Technology, p. 2. Beijing Institute of Technology Press, Beijing (2018)
8. Wang, X., et al.: Chang'e-1 Satellite Control Sub-system ground test system design. Spacecraft Eng. **3** (2008)