Research on Fault-Tolerant Control System for Space Modular Manipulator System

YE Ping[1], SUN Han-xu[2], JIA Qing-xuan[2], WANG Xin-sheng[2]
(*1.Robotics Research Institute, Beijing University of Aeronautics and Astronautics, Beijing 100083, China*)
(*2. Automation School, Beijing University of Posts and Telecommunications, Beijing 100876, China*)

Abstract: This paper studies a fault-tolerant control system for a space modular manipulator system mounted on space station or other spacecrafts such as satellites, located in low earth orbit. Design technologies for traditional industrial manipulator systems cannot be directly used to the space ones due to the special space environment and compactness. Considering the extremely tight constraints on mass, power consumption, volume, cost and "design-to-orbit" schedules, the fault-tolerant control system is developed mainly based on commercial-off-the-shaft components. The features of the hardware and software of the fault-tolerant control system are presented. The performance specifications are also discussed. Because many space proven design technologies and experiences are adopted, the fault-tolerant control system is characterized by high reliability and practicability.

Key words: fault-tolerant control system; space modular manipulator system; commercial-off-the-shaft(COTS)

空间模块化机械手系统容错控制系统研究．叶 平，孙汉旭，贾庆轩，王新升.中国航空学报(英文版)，2006，19(3)：260-265.

摘 要：研究搭载在空间站或其他近地轨道航天器上的空间模块化机械手系统的容错控制系统。与地面应用环境相比较，空间特殊的应用环境存在许多不同，因而传统工业机械手系统的设计技术不能直接应用于空间机械手系统的设计。考虑到空间应用对系统质量、体积、功耗、研制费用和周期所提出的苛刻要求，主要采用 COTS 器件设计并实现该空间机械手系统的容错控制系统，介绍了系统的软硬件体系结构及其特点，最后给出系统的性能参数。在系统设计中采用许多已有的设计技巧与工程经验，具有高可靠性和工程实用性。

关键词：容错控制系统；模块化空间机械手系统；COTS 技术
文章编号：1000-9361 (2006)03-0260-06 **中图分类号：**TP302.8 **文献标识码：**A

Space manipulator systems are necessary for future space missions to construct, maintain and repair spacecrafts in orbits, and reduce the need for costly and hazardous Extra-Vehicular Activity[1,2]. Hence, it is an important subject to develop a space manipulator system, which can be mounted on space station or satellites, such as Canada's SSRMS[3], Germany's ROKVISS[4], Japan's JEMRMS[5] and so on.

Due to the special space environment and compactness, the design technologies for traditional industrial manipulator systems cannot be directly used to the space ones, and space technologies often lag considerably that in common household use on the ground[6].

Fault-Tolerant Control System (FTCS) is the brain of space manipulator systems. Missions of this sophisticated piece of equipment include status

Recived date:2005-08-03; Revision received date:2005-11-25
Foundation Item: "863" National High Technology Foundation in Astronautics(2005AA742030)

monitoring, fault-tolerance, resource reconfiguration and communication. Therefore, a high-performance FTCS is required to handle its various requirements.

Advances in VLSI microelectronics have catalyzed a new species of low-cost yet sophisticated and highly capable spacecrafts, which have been built and launched successfully mainly comprised of Commercial-Off-The-Shelf (COTS) components[7-9]. Practical experiences have shown that COTS components are particularly attractive for these spacecrafts located in low earth orbit, due to their relatively lower power consumption, more powerful processing performance and easier availability as compared with those radian hardened components[10].

The work on a FTCS of a Space Modular Manipulator System (SMMS) is summarized. The FTCS is developed mainly based on COTS components.

This paper is organized as follows. In Section l, the hardware architecture is described. The software architecture is presented in Section 2. In Section 3, the excellent performance specification of the FTCS is summarized. Section 4 is the conclusions.

1 Hardware Architecture of the FTCS

The FTCS is a distributed control system. Its communication architecture is based on a dual-redundant power bus, a dual-redundant external Controller Aera Network (CAN) bus and a dual-redundant internal CAN bus. It communicates with the mounted spacecraft by the external CAN bus; the internal CAN bus is used to transmit command and data inside the SMMS. The FTCS is composed of the following nodes (see Fig.1):

① the Central Control Unit (CCU);

② one Dual-joint Control Unit (DCU)for each two adjoining revolute joints;

③ the End-effector Control Unit (ECU) for the end-effector.

Utilizing a modular design approach, each control unit is incorporated into the corresponding mechanism of the SMMS, except the CCU installed

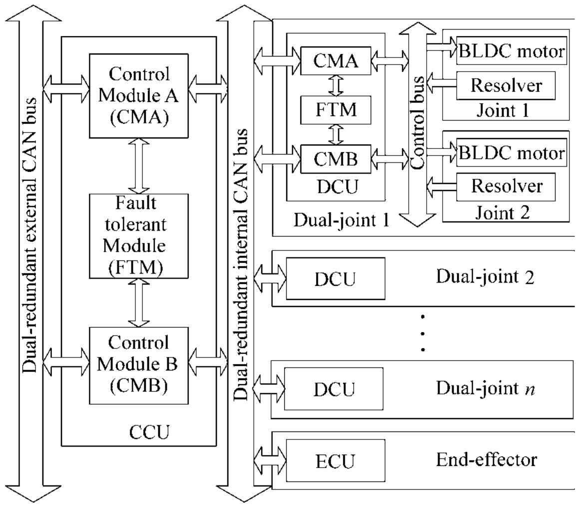

Fig.1 Hardware architecture of the FTCS

inside the mounted spacecraft. This distributed architecture, having the control system located close to the sensors and the actuators[11], eliminates the need for multi-wire cable harnesses, thereby saving system mass and increasing system reliability, and improves robustness with respect to EMC.

Candidate data buses for space application can be CAN, RS485/RS422, LVDS or MIL-STD-1553B. A trade-off analysis has been conducted, indicating that the CAN bus should be the preferred choice, owing to its greater advantage in terms of hardware simplicity, permitting significant mass and dimension savings, and providing a communication bandwidth that satisfies the requirements.

Considering the extremely tight constraints, the CCU, the DCU and the ECU are designed, mainly composed of COTS components. The designed CCU is based on high-performance ATMEL AT91x40 serial RISC ARM microcontroller, while the DCU and the ECU are based on high-performance TI C24x Digital Signal Processor (DSP). Each is designed to be dual modular redundant architecture, namely, having two independent and identical control modules, one active and the other standby. When the active control module fails, the standby control module will be activated by decisions of fault-tolerant module, and supersede the failed one. Thus, the high reliability of the FTCS is ensured by hardware redundancy at system level.

1.1 Hardware architecture of the CCU

The detailed architecture of the control module of the CCU is shown in Fig.2. It is an ARM7DTMI-based data handling computer system with Error Detection and Correction (EDAC) protected program and data memory, including two dual-redundant CAN bus interfaces. This microcontroller incorporates the ARM7TDMI processor core, features 1M-byte of on-chip Flash memory in a monolithic chip, and gives about 40 MIPS at 48 MHz internal frequency.

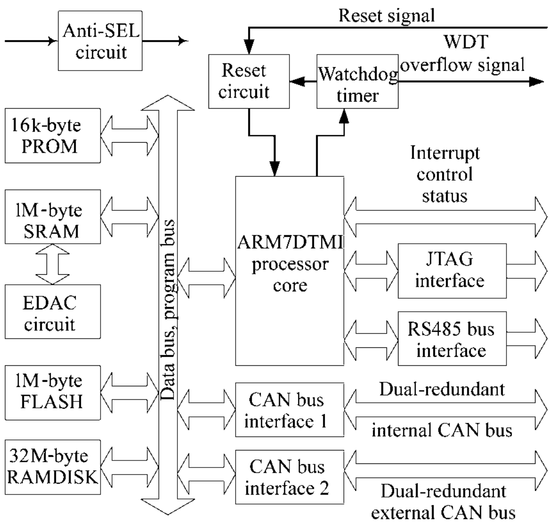

Fig.2 Block diagram of the control module of the CCU

The CCU memory system consists of PROM, Flash, SRAM and RAMDISK. The 16k-byte PROM contains two copies of bootstrap codes in two separate PROM pages. The 1M-byte Flash contains two copies of OS kernel and application software, which will be loaded into the SRAM memory for execution during the run time. The 1M-byte SRAM managed by OS kernel is the main memory. A hardware EDAC circuit is implemented to correct one bit and detect two bit errors using hamming code. The 32M-byte RAMDISK is used for data storage. Instead of hardware EDAC circuit, RAMDISK uses a block coding similar to Reed-Solomon code to correct one byte error from 252 bytes data block adding three bytes codes to the data block.

Two CAN bus interfaces are implemented, one for dual-redundant external CAN bus, the other for dual-redundant internal CAN bus. A RS485 bus interface is designed for communication between the two control modules of the CCU. An on-chip Watchdog Timer (WDT) is programmed to monitor the hardware and software operations. It will generate a reset signal to processor and restart the control module, if it is not periodically serviced by software by having the correct key written. A JTAG interface is also implemented for debugging during software development. An anti-SEL circuit is used to limit the current of the control module whenever Single-Event Latch-up (SEL) occurs.

1.2 Hardware architecture of the DCU and the ECU

The DCU and the ECU have the same design, except for motion control interface. The former is designed to drive two BLDC motors of two adjoining revolute joints, while the latter only one BLDC motor. The detailed architecture of the control module of the DCU is shown in Fig.3.

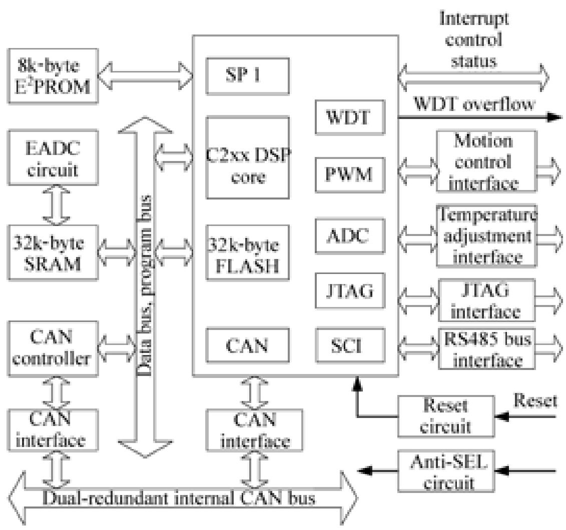

Fig. 3 Block diagram of the control module of the DCU

The DCU is a TI C24x DSP-based servo controller. It features EDAC protected memory, a dual-redundant CAN bus interface, interfaces for motion control and so on. TI C24x DSP is part of the TMS320C2000™ platform fixed-point DSP, offers 32k-byte on-chip Flash memory. Several advanced peripherals, optimized for digital motor and motion control application, are integrated to provide a true single-chip DSP controller, such as PWM generator, analog-to-digital converter, CAN module and so on. Its increased processing performance and a higher level of peripheral integration make it an excellent candidate for motion control and data processing applications, such as the DCU and the ECU, which have very severe power, mass and volume constraints, because they are incorporated into the corresponding mechanism of the SMMS separately.

Instead of using PROM, the bootstrap code is stored into 8k-byte serial E^2PROM which features a SPI interface. The E^2PROM is also used as a nonvolatile memory to store Control Word and history. The application software is stored into the on-chip Flash memory, for the sake of mass and volume savings. The 32k-byte EDAC protected SRAM is managed by OS kernel. This DSP offers a CAN module with CAN 2.0B protocol. An extended CAN bus controller is designed to interface the dual-redundant internal CAN bus.

A motion control interface is designed to drive BLDC motors, including PWM generator, shaft angle measurement and current sensing circuits.

A temperature adjustment interface is implemented to sense and adjust the temperature inside each dual-joint or the end-effector.

Similarly to the CCU, a RS-485 interface, a JTAG interface, an anti-SEL circuit and an on-chip WDT are implemented also.

2 Software Architecture of the FTCS

The FTCS software is the soul of the FTCS. The FTCS software is composed of the following:

① the Central Control Software (CCS), running on the CCU;

② the Dual-joint Control Software (DCS), running on each DCU;

③ the End-effector Control Software (ECS), running on the ECU.

The CCS performs high level operation of the manipulator of the SMMS, such as path planning, inverse kinematics, dynamic coordinated control of multiple axes and so on; while the CCS is in charge of status monitor, fault-tolerance, resource management and configuration, communication, and program unloading and updating.

The main missions of the DCS include motion control, temperature adjustment, fault-tolerance, resource management and configuration inside the dual-joint, and communication between the DCU and the CCU.

The ECS provides grasp and release operation of the end-effector.

The CCS software consists of several tasks that run concurrently under the control of the multitasking kernel VxWorks. VxWorks is a popular Real-Time Operating System (RTOS) used in aerospace field, while the DCS and the ECS run under the uC/OS-II, a free source-open RTOS instead of the VxWorks, because there is unavailable of VxWorks on the TMS320C2000™ platform and the DCS and the ECS are relatively simple contrasted with the CCS.

The software architectures of the CCS, the DCS and the ECS are the same. The only difference is the RTOS. The detailed software architecture is shown in Fig.4.

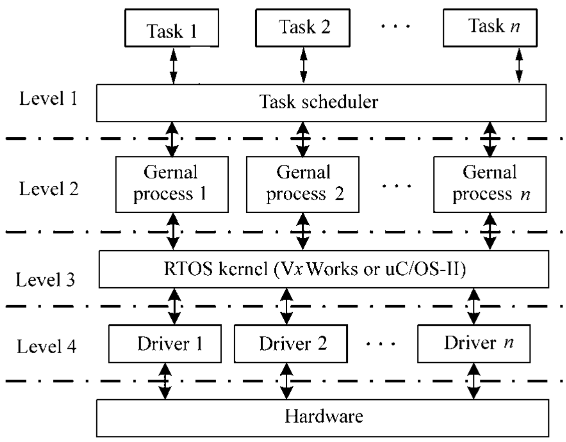

Fig. 4 Software architecture of CCS, DCS and ECS

At the top-level, application software for specific in-orbit experimental tasks and a task scheduler process are executed.

General high-level processes are at the hypo-top level, including CAN bus high protocol, fault-tolerance, resource management and reconfiguration, and various algorithms required by the operation of the SMMS, such as path planning and inverse kinematics algorithms of the CCS, PID algorithms for motion control of the DCS and the ECS.

The RTOS kernel is at the level 3 in the architecture, VxWorks or uC/OS-II. The RTOS kernel performs system level functions such as task scheduling, task switching and inter-task communication.

At the level 4, it is the driver for various interfaces used in the FTCS, such as CAN bus, RS485 bus, etc.

The FTCS software has been designed around a message-driven model. When a message is received from CAN bus or RS485 bus, the task scheduler creates and activates the corresponding tasks. When a task is completed, the task scheduler kills it and returns the execution results.

On launch, the FTCS software has only a small bootstrap code running from PROM or E^2PROM, and for them to be generally useful, a suite of application softwares must be loaded into SRAM from Flash memory by the bootstrap code. When a fault is detected in application software, the bootstrap code allows uploading a new copy of the FTCS software by telecommand from the ground. Additionally, a new version of the FTCS software with new missions can be uploaded for new in-orbit experiments.

3 Performance Specifications of the FTCS

The FTCS of the SMMS is developed mainly based on COTS components with high-performance processing capabilities and high-level integration of peripherals. Table 1 and Table 2 show the main performance specifications of the CCU and the DCU (the ECU) respectively.

Table 1 Performance specifications of the CCU

Characteristic	Performance
Throughput	32bit RISC processor, 40MIPS@ 48MHz
Memory	16k-byte PROM 1M-byte SRAM with EDAC 1M-byte Flash 32M-byte RAMDISK
Dimensions	257mm×194mm×120 mm
Mass	Less than 4 kg
Power consumption	Less than 4 W
Supply voltage	+28 V
CAN bus	512k-bit/s
RS485 bus	19.2k-bit/s

Table 2 Performance specifications of the DCU (the ECU)

Characteristic	Performance
Throughput	16bit fixed-point DSP near 20MIPS@ 20MHz
Memory	8k-bytes E^2PROM 32k-bytes SRAM with EDAC 32k-bytes Flash
Dimensions	100mm×90mm×62 (40) mm
Mass	Less than 0.8(0.6)kg
Power consumption	Less than 6(4) W
Supply voltage	+28 V, +15 V and +5 V
CAN bus	512k-bit/s
RS485 bus	19.2k-bit/s

Note: () are for the ECU.

4 Conclusions

In this paper, the design of the FTCS and its software development are presented. The design of the FTCS is driven by the extremely tight constraints of small mass, low power consumption, low volume, stringent cost control and fast “design-to-orbit” schedules. The developed FTCS is based on AT91x40 serial RISC ARM microcontroller and TI C24x DSP. The VxWorks real-time multi-tasking operating system is used for the CCU and uC/OS-II is used for the DCU and the

ECU. The performance of the FTCS satisfies the requirements so far. The research on the design of the FTCS for the SMMS will be continued and their capability will be further proved in near future.

References

[1] Senda K, Mano S, Matsumoto T, et al. A study toward an autonomous space robot [C]//Proceedings of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space. Japan：NARA, 2003.

[2] Schuler S, Hirt M, Schmidt M, et al. Dextrous robot arm [C]//Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk, Netherlands: ESTEC, 2004.

[3] Piedboeuf J C, Dupuis É. Recent canadian activities in space automation and robotics—an overview [C]//Proceeding of the 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space. Japan：NARA, 2003.

[4] Hirzinger G, Brunner B, Lampariello R, et al. Advances in orbital robotics[C]//IEEE Int Conf on Robotics and Automation. San Francisco, CA, 2000: 898-907.

[5] Iwata T. Current status of space robotics activities in Asia [C]//Proceeding of the 7th International Symposium on Artificial intelligence, robotics and automation in space. Japan: NARA, 2003.

[6] Sweeting M N. Cost-effective microsatellites for low earth orbit communications [C]//Singapore ICCS '94. 1994: 400-404.

[7] Underwood C I, Oldfield M K. Observed radiation-induced degradation of commercial-off-the-shelf (COTS) devices operating in low-earth orbit [J]. IEEE Transactions on Nuclear Science, 1998, 45(6)：2737-2744.

[8] Underwood C I. A low cost modular COTS-based nano-satellite-design, construction, launch and early operations phase [C]// Proceedings of the 15th Annual AIAA/USU Conference on Small Satellites. SSC01-VI-7, AIAA/USU, 2001.

[9] Underwood C I, Crawford M, Ward J W. A low cost modular nanosatellite based on commercial technology [C]//Proceedings of the 12th Annual AIAA/USU Conference on Small Satellites. SSC98-V-4, AIAA/USU, 1998.

[10] McLoughlin I V, Gupta V, Sandhu G S, et al. Fault tolerance through redundant COTS components for satellite processing applications[C]//Information, Communications and Signal Processing the Fourth Pacific Rim Conference on Multimedia. 2003: 296-299.

[11] Rusconi A, Magnani P G, Grasso T. DEXARM – a dextrous robot arm for space applications [C]//Proceedings of the 8th ESA Workshop on Advanced Space Technologies for Robotics and Automation. Noordwijk Netherlands: ESTEC, 2004.

Biography:

YE Ping Born in 1979, he received B.S. from Beijing University of Aeronautics and Astronautics in 2001, and now he is a PH.D candidate in Robotics Research Institute, Beijing University of Aeronautics and Astronautics. He does cooperative research in Space Robotics Research Institute of Automation School, Beijing University of Posts and Telecommunications. His current research interest is fault-tolerant control of space modular manipulator system. Tel: (010) 62282571; E-mail: bronieye@163.com