

基于实时操作系统的故障诊断 高可靠测控系统设计

解梦迪, 姜姗姗, 郭照峰, 贾凡

(北京航天测控技术有限公司, 北京 100041)

摘要: 针对某型地面测控系统连续一年加电的高可靠性设计需求, 目前所采用的双机冗余方案缺少对系统软硬件状态异常因素的考虑, 系统在长期运行过程中存在系统局部异常而无法触发双机切换从而导致冗余失效的可能性, 这些异常主要包括任务 STOP 异常、CPU 温度过高、CPU 占用率过大、磁盘剩余空间过小以及 IO 状态异常等; 针对这些问题, 提出一种系统故障诊断与容错方法, 对系统故障源进行了综合研究和分析; 文章采用任务实时监测、最小二乘法以及哈希算法等关键技术和方法实现系统故障诊断与容错处理, 经实际型号项目验证了一年连续加电工作正常, 设计满足了系统对高可靠性需求的应用需求。

关键词: 高可靠性; 测控系统; 故障诊断; 容错处理; 实时系统

Design of High Reliable Measurement and Control System for Fault Diagnosis Based on RTOS

XIE Mengdi, JIANG Shanshan, GUO Zhaofeng, JIA Fan

(Beijing Aerospace Measurement & Control Technology Co., Ltd., Beijing 100041, China)

Abstract: Aiming at the high reliability design requirements of a certain type of ground measurement and control system that has been powered up for one year continuously, dual redundancy scheme lacks the consideration of abnormal factors of system software and hardware status at present. During the long-term operation of the system, there are system local exceptions that can't trigger the dual machine switch, leading to the possibility of redundancy failure. These exceptions mainly include the task STOP exception, excessive CPU temperature, excessive CPU utilization, insufficient disk space, and abnormal IO status. For this problem, a system fault diagnosis and fault tolerance method is proposed to comprehensively study and analyze the system fault sources. Key technologies and methods such as task real-time monitoring, least square method and Hash algorithm are used to realize the system fault diagnosis and fault tolerance processing, the actual model project has verified the normal operation of continuous power on for one year, and meets the high reliable applications and requirements in the system.

Keywords: high reliability; test and control system; fault diagnosis; fault tolerant; RTOS (real time operating system)

0 引言

目前在地面装备中为提高系统长期加电工作可靠性, 双机冗余系统已经广泛应用, 但在系统软件容错方面存在着一些不足, 会因为系统测控软件的问题导致双机冗余失效, 进而导致双机系统没有起到作用。为解决因系统测控软件异常导致的冗余失效问题, 提出一种任务状态、CPU 温度、CPU 占用率^[1]、磁盘剩余空间以及 IO 状态监测的故障诊断与容错方法, 该方法进一步提高系统稳定性和可靠性, 保障测控设备长期连续加电的需求。

1 系统结构及原理

系统采用基于龙芯 CPU、CPCI 总线以及实时操作系统的组合架构, 主要实现测控流程、数字量 IO 的输入输出控制、AD/DA 模拟量采集与输出、网络通讯、串口通讯以及热备冗余等功能。如图 1 所示, 为实现双机冗余^[2]功能开发

的冗余控制模块, 它是嵌入式测控系统双机冗余控制和切换的核心模块, 该模块产生双机通信信息, 并通过对机的信息判断是否需要本机切换成主机或从机状态, 从而给 CPU 主从机切换指令, 实现主从机的热备冗余切换, 保障双机系统整体可靠性。

如图 1 所示, 双机采用冗余控制板构建双机冗余架构, 其中 1 号机的 sd1 和 sd2 是冗余控制板发送的两路心跳信号, rc1 和 rc2 是冗余控制板接收的对机两路心跳信号; 同样, 2 号机也有两路心跳信号发送和两路心跳信号接收, 如图 2 所示。

而系统故障如任务异常、CPU 温度过高、CPU 占用率过大、磁盘剩余空间不足以及 IO 状态异常等因素没有被考虑到容错机制中, 文章将基于上述双机系统开展对系统故障因素的影响研究并提出一种监测与容错方案设计与实现。

收稿日期: 2022-10-20; 修回日期: 2022-11-14。

作者简介: 解梦迪(1989-), 男, 湖北荆门市人, 硕士, 高级工程师, 主要从事测控实时测试技术方向的研究。

引用格式: 解梦迪, 姜姗姗, 郭照峰, 等. 基于实时操作系统的故障诊断高可靠测控系统设计[J]. 计算机测量与控制, 2023, 31(2): 8-14, 20.

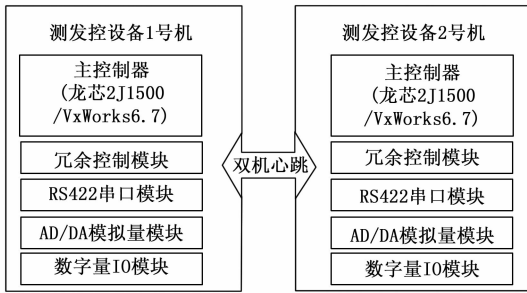


图 1 系统结构图

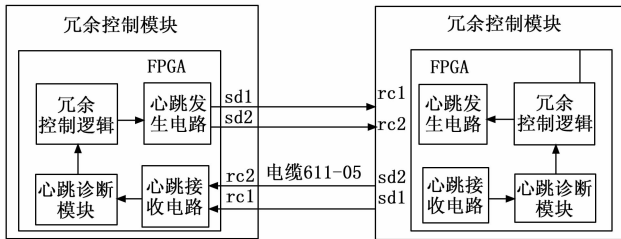


图 2 双机心跳检测结构图

2 系统硬件设计

系统主要包括龙芯微处理器 (MCU, micro controller unit) 模块、多种外设功能模块 (冗余控制模块、数字量 IO 模块、AD/DA 模拟量模块、RS422 串口模块以及供电模块等)。

2.1 龙芯 MCU 模块

2J1500+1A500_3U_CPCI 板卡是 3U 计算机模块, 采用龙芯 2J1500 处理器, 实现符合 Compact PCI 2.0R3.0 规范的 3U Compact PCI 接口计算机模块, 控制器原理如图 3 所示。该计算机模块具有完整的计算机功能, 集成网络、

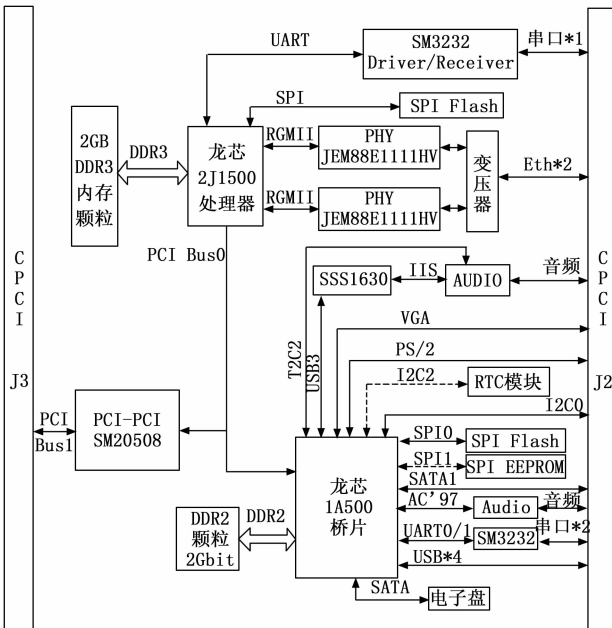


图 3 龙芯控制器原理图

显示、PCI-PCI 桥、SATA 接口、声卡、串口、VGA、USB 接口等功能。同时, 为避免因静电造成的板卡 IC 器件损坏, 故针对各个接口如 USB、SATA、PS/2、VGA 做了特定的 ESD 防护设计, 将 ESD 器件放在连接器侧放置, 同时滤波器件紧挨 ESD 器件, 最大程度地保护因静电引起的瞬态电流而对板卡器件造成的损坏, 延长板卡器件使用周期。

2.2 冗余控制模块

冗余控制模块是基于 CPCI 总线和 FPGA 的高速逻辑控制模块, 如图 4 所示。该模块用于完成包括本机切换控制、报警控制、隔离控制等功能。硬件功能组成包括 RS422 隔离通信、继电器驱动、LED 驱动、电源管理、电平转换、电源监控、CPCI 总线及 FPGA 等电路。

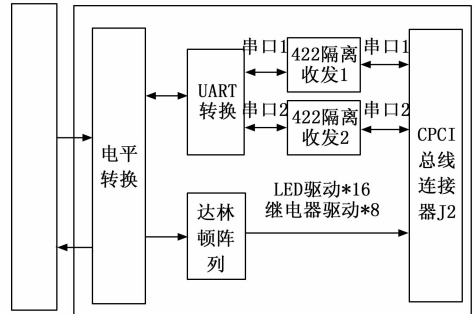


图 4 冗余控制模块原理图

2.3 数字量 IO 模块

数字 IO 模块输出 TTL 电平信号, 用于测控组合设备前面板信号指示灯控制, 同时采集来自串口通信切换模块的状态回采信号用于控制器模块对设备隔离状态自检回采。控制器模块通过数字 IO 模块驱动程序控制指示灯驱动电路实现面板指示灯状态控制, 原理如图 5 所示。

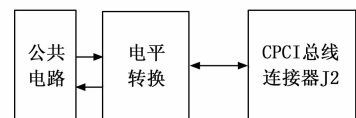


图 5 数字量 IO 模块原理图

2.4 AD/DA 模拟量模块

AD/DA 模块为基于 CPCI 总线和 FPGA 的高速逻辑控制模块, 用于完成包括模拟量电压采集、模拟电压输出和开关量 IO 等功能。硬件功能组成包括 AD、DA、IO、电源管理、CPCI 总线、FPGA 等电路, 原理如图 6 所示。

2.5 RS422 串口模块

RS422 串口模块是基于 CPCI 总线的 4 路隔离串行通信接口模块, 如图 7 所示。422 串口模块主要基于 UART 模块来实现串行通信。主要由 UART 模块、电平转换模块、422 隔离收发模块、电源管理、CPCI 总线、FPGA 等电路组成。

2.6 电源模块

电源模块主要功能将输入直流 DC28V 通过内部极性保

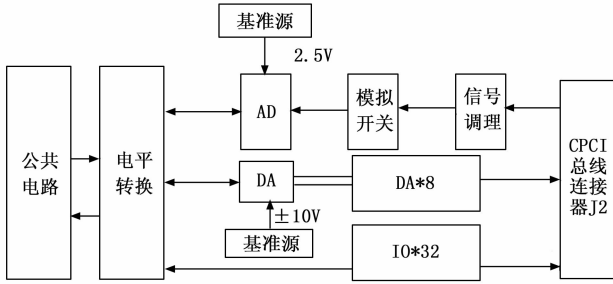


图 6 AD/DA 模拟量模块原理图

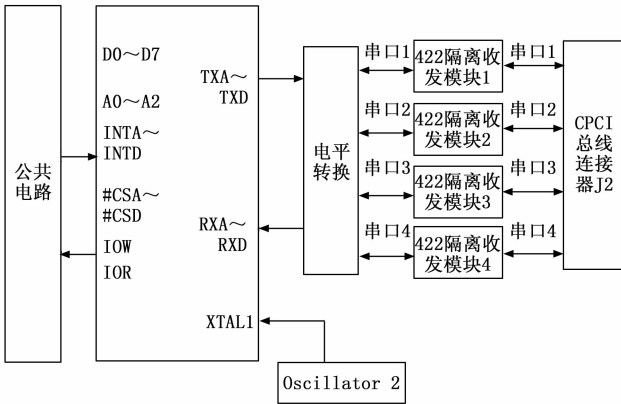


图 7 RS422 串口模块原理图

护组件、滤波器组件、输入滤波电路、DC/DC 变换器组件及输出滤波组件变换一组稳定的直流输出，给系统供电，原理如图 8 所示。

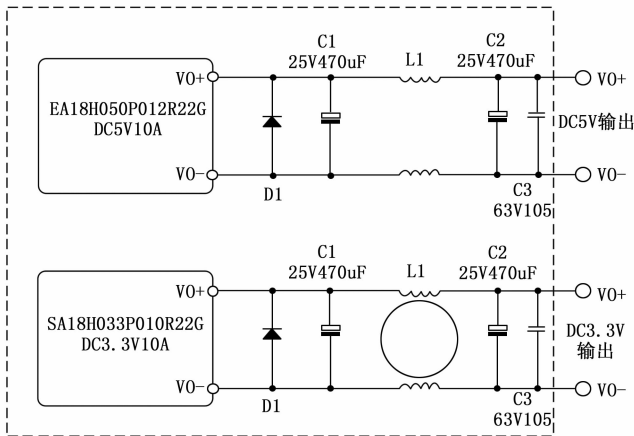


图 8 电源模块原理图

3 系统测控软件设计

系统测控软件运行在测控设备中，实现自检、网络接收指令、控制指令执行、数据采集与控制、组播数据回传、灾备恢复、应用故障诊断等。文章详细对故障信息采集以及故障诊断设计进行阐述。

文章主要分析任务监测^[3]状态、CPU 温度、CPU 占用率、磁盘剩余空间状态以及 IO 状态故障诊断分析，根据故

障分类形成一组结果存储在一维数组中，从数组中查找找到并定位故障，而顺序查找和二分查找的实践复杂度分别是 $O(n)$ 和 $O(\log n)$ ，对于实时系统选择用复杂度 $O(1)$ 的查找算法更加合适。而哈希算法是一种复杂度 $O(1)$ 的查找算法，其中位图是其中一种应用，本文将采用位图查找故障元素。

故障因素按照数组设计为 $element_array[] = \{ \text{“任务状态”}, \text{“CPU 温度”}, \text{“CPU 占用率”}, \text{“磁盘剩余空间”}, \text{“IO 状态”} \} = \{4, 3, 2, 1, 0\}$ ，根据 $element_array$ 和切换结果保存数组 $List_switch$ 生成位图映射表 BitMap，如表 1 所示。

表 1 位图映射表 BitMap

BitMap 位置	结果
0	normal
1	IO _{abnormal}
2	VHDC _{abnormal}
3	IO _{abnormal} && VHDC _{abnormal}
4	RCPU _{abnormal}
5	IO _{abnormal} && RCPUs _{abnormal}
6	IO _{abnormal} && VHDC _{abnormal}
7	IO _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal}
8	TCPU _{abnormal}
9	TCPU _{abnormal} && IO _{abnormal}
10	TCPU _{abnormal} && VHDC _{abnormal}
11	TCPU _{abnormal} && VHDC _{abnormal} && IO _{abnormal}
12	TCPU _{abnormal} && RCPUs _{abnormal}
13	TCPU _{abnormal} && RCPUs _{abnormal} && IO _{abnormal}
14	TCPU _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal}
15	TCPU _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal} && IO _{abnormal}
16	TASK _{abnormal}
17	TASK _{abnormal} && IO _{abnormal}
18	TASK _{abnormal} && VHDC _{abnormal}
19	TASK _{abnormal} && VHDC _{abnormal} && IO _{abnormal}
20	TASK _{abnormal} && RCPUs _{abnormal}
21	TASK _{abnormal} && RCPUs _{abnormal} && IO _{abnormal}
22	TASK _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal}
23	TASK _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal} && IO _{abnormal}
24	TASK _{abnormal} && TCPUs _{abnormal}
25	TASK _{abnormal} && TCPUs _{abnormal} && IO _{abnormal}
26	TASK _{abnormal} && TCPUs _{abnormal} && VHDC _{abnormal}
27	TASK _{abnormal} && TCPUs _{abnormal} && VHDC _{abnormal} && IO _{abnormal}
28	TASK _{abnormal} && TCPUs _{abnormal} && RCPUs _{abnormal}
29	TASK _{abnormal} && TCPUs _{abnormal} && RCPUs _{abnormal} && IO _{abnormal}
30	TASK _{abnormal} && TCPUs _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal}
31	TASK _{abnormal} && TCPUs _{abnormal} && RCPUs _{abnormal} && VHDC _{abnormal} && IO _{abnormal}

其中: normal 表示无异常; IO_{abnormal} 表示 IO 异常; VHDC_{abnormal} 表示硬盘剩余异常; RCPUs_{abnormal} 表示 CPU 占用率异常; TCPUs_{abnormal} 表示 CPU 温度异常; TASK_{abnormal} 表示任务异常。

图 9 $List_{switch}$ 中各状态位置排列, D4 表示任务状态、D3 表示 CPU 温度状态、D2 表示 CPU 占用率状态、D1 表示磁盘状态、D0 表示 IO 操作状态, 同理可以处理警告保存数组 $List_{warn}$ 。

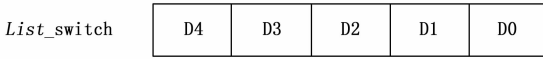


图 9 数组示意

通过维护记录位图映射表 BitMap 和故障因素的数组 $List_{switch}$, 通过计算数组 $List_{switch}$ 所表示的值, 然后索引映射表 BitMap 就可以找到当前哪些因素触发切换条件。这个数组的值表示的位置和映射表 BitMap 中的位置是一一对应的, 可以快速定位当前系统故障因素。

3.1 任务状态采集监测程序设计

任务状态表示实时系统中任务运行状态, 有多种状态如 “STOP”、“DELAY”、“SUSPEND” 以及 “READY” 等。

测控软件启动时, 测控软件建立并启动任务时, 有两类获取到任务 ID 的途径。一类是将其返回的任务 ID 实时的保存在任务 ID 列表中。另一类是系统任务, 通过系统函数接口使用名称查找其任务 ID, 并将其存到任务 ID 列表中, 程序动态维护任务 ID 列表。

设计注册所需要监测的任务详细见表 2, 任务状态监测逻辑如图 10 所示。

表 2 监测任务列表

序号	任务代号	任务名称	备注
1	RECV_XINGKONGDATA	网络处理任务	
2	RECV_XINGKONGDATA	箱压接收与处理任务	
3	RECV_XINGKONGDATA	增压流程任务	
4	Net0	系统网络底层任务	
5	HEARTBEAT_TASK	监测对机心跳和判读任务	

任务状态采集与监测设计概述如下:

1) 启动任务时记录任务返回值为任务的 ID 值并依次存入到固定数组 $idarray [n]$ 。

2) 动态任务启动和退出主要包含删除任务前更新数组 $idarray [n]$ 的 ID 和正常退出前更新数组 $idarray [n]$ 的 ID。

3) 针对数组 $idarray [n]$ 进行扫描, 判断其任务状态, 识别到任务 “STOP” 异常后, 完成异常任务切换标志设置; 识别到任务 “SUSPEND”, 设置报警设置, 表明系统出现任务 SUSPEND 的状态, 记录日志。

动态任务的个数有限, 可以设计固定任务在数组前 N 位, 动态任务的 ID 存入固定的后半段 $N-1 \sim M$, 系统任务存入 $M+1 \sim N$, 逻辑设计如图 10 所示。

在任务状态采集与监测模块中, 完成任务 ID 维护、任务状态实时监测以及监测结果标记等工作, 为故障诊断模块提供处理依据。

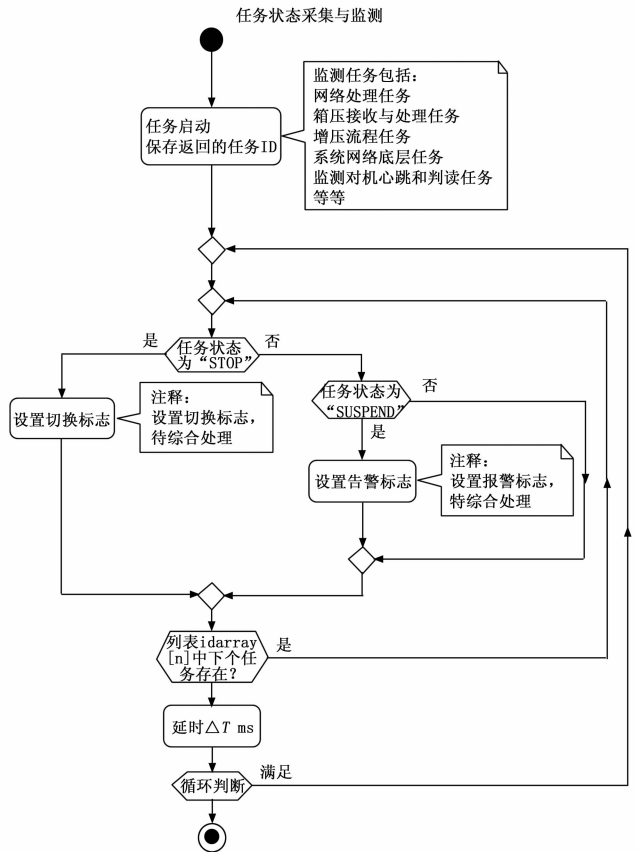


图 10 任务状态获取与监测逻辑设计图

表 3 系统运行任务状态监测判据详细列表

序号	监测内容	报警门限	切换门限	异常可能原因
1	任务状态	SUSPEND	STOP	内存访问越界、操作系统崩溃等

3.2 CPU 温度、CPU 占用率以及磁盘剩余空间监测采集程序设计

软件设计独立监测模块^[4], 周期性的采集 CPU 温度 (T_{CPU})、CPU 占用率 (R_{CPU}) 以及磁盘剩余空间 (V_{HDC}) 信息, CPU 温度/占用率以及磁盘空间状态判据如表 4 所示, 传递给信息处理模块使用, 如图 11 所示。

表 4 CPU 温度/占用率以及磁盘监测判据详细列表

序号	监测内容	报警门限	切换门限	异常可能原因
2	CPU 温度	$T_{CPU} \geq 70^{\circ}C$	$T_{CPU} \geq 85^{\circ}C$	机箱散热不良、风扇故障、CPU 自身损坏等
3	CPU 占用率	$R_{CPU} \geq 50\%$	$R_{CPU} \geq 80\%$	程序跑飞、操作系统崩溃等
4	磁盘剩余空间	$V_{HDC} \leq 3 \text{ GB}$	$V_{HDC} \leq 1 \text{ GB}$	日志写满、程序跑飞、操作系统崩溃等

软件对于系统的运行状态 (CPU 温度、CPU 占用率以及磁盘剩余空间) 故障进行 “故障趋势分析”, 即当表 4 中的检测项目的采集值超过 “报警门限” 或 “切换门限” 后, 软件会对之后 5 个检测周期内该参数的变化趋势进行预测

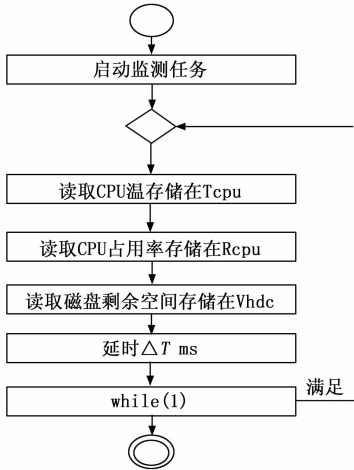


图 11 CPU 温度/占用率/磁盘剩余空间监测逻辑

与判断，根据其曲线走向执行相应的报警动作或切换动作，防止指标出现极短时间超过阈值后回落到正常范围情况时出现虚报，从而降低虚警概率。

系统监测过程中，被监测信息（CPU 温度、CPU 占用率以及磁盘剩余空间的变化规律随时间是不断变化的，为了从监测数据中提取出有效信息，需要对监测数据的走势做出假设。

本文采用最小二乘法^[5]实现对严重故障的预测与分析，最小二乘法是一种数据优化技术，它通过最小化误差的平方和寻找数据的最佳函数匹配。利用最小二乘法可以简便的预测未来的数据，并使得这些求得的数据与实际数据之间误差的平方和为最小。

具体过程如下：“报警门限”和“切换门限”可以通过用户配置文件设置。当采样值超过“报警门限”或者“切换门限”时，采样过去 5 个周期的实测值采用线性拟合的方法拟合出曲线， $f(t) = a + b * t + \dots + c * t^n$ ， t 表示时刻， $f(t)$ 表示 t 时刻的拟合值，根据此公式预测未来趋势，进而来预测超过“报警门限”或“切换门限”是否有效发生，完成“报警”或“切换”标志设置，逻辑设计如图 12 所示。

3.3 IO 模块状态

系统监测 IO 输出与采集的结果 IO_{out} 和 IO_{in} 比对，设计当结果 $IO_{out} \cong IO_{in}$ ，判定 IO 控制异常，设置切换状态。

4 故障诊断设计

综合任务状态、CPU 温度、CPU 占用率、磁盘剩余空间以及 IO 状态，采用位图^[6-7]查找算法，快速检测出当前系统中需要切换的因素。

故障诊断^[8-22]实时监测任务监测状态、CPU 温度、CPU 占用率以、磁盘剩余空间状态以及 IO 状态等，并根据各项因素保存的“警告”列表 $List_{warn}$ 和“切换”列表 $List_{switch}$ 中状态，实时识别并做出操作。

识别是 normal 状态不做处理；识别是 warn 状态记录日志提示；识别是 switch 状态执行停止喂狗并记录日志。

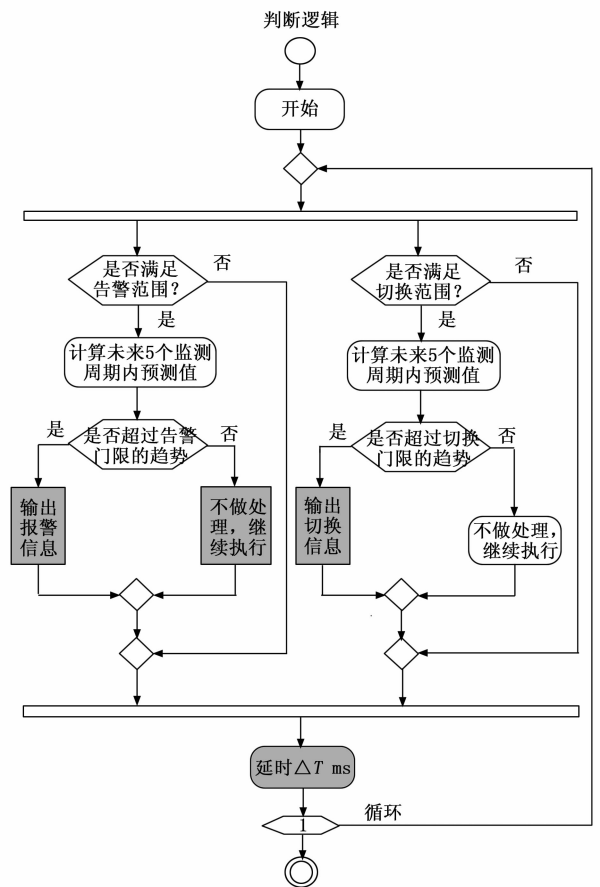


图 12 CPU 温度/占用率以及磁盘空间判断逻辑图

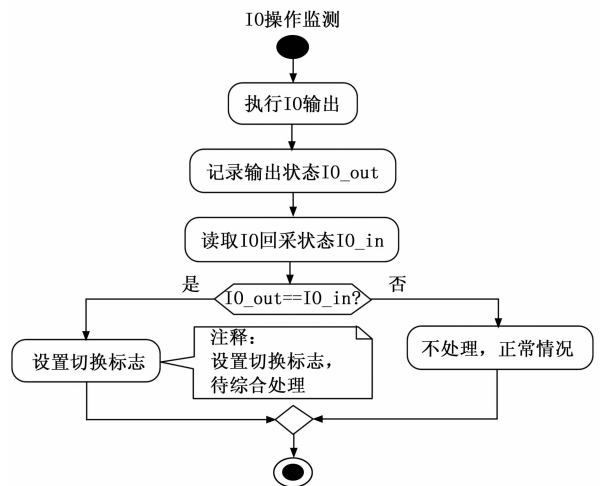


图 13 IO 操作监测逻辑

综合判断系统报警与切换条件如表 5 所示。

表 5 综合判断表

序号	处理	条件内容
1	正常	五种状态均正常
2	报警	五种状态无切换异常, 存在任意报警状态
3	切换	监测任一任务异常 T_{CPU} 异常 R_{CPU} 异常 V_{HDC} 异常 IO 异常

任务状态、CPU 温度、CPU 占用率以及磁盘剩余空间状态独立判断, 形成报警和异常切换结果, 依据表逻辑, 综合判断, 设计如图 14 所示。

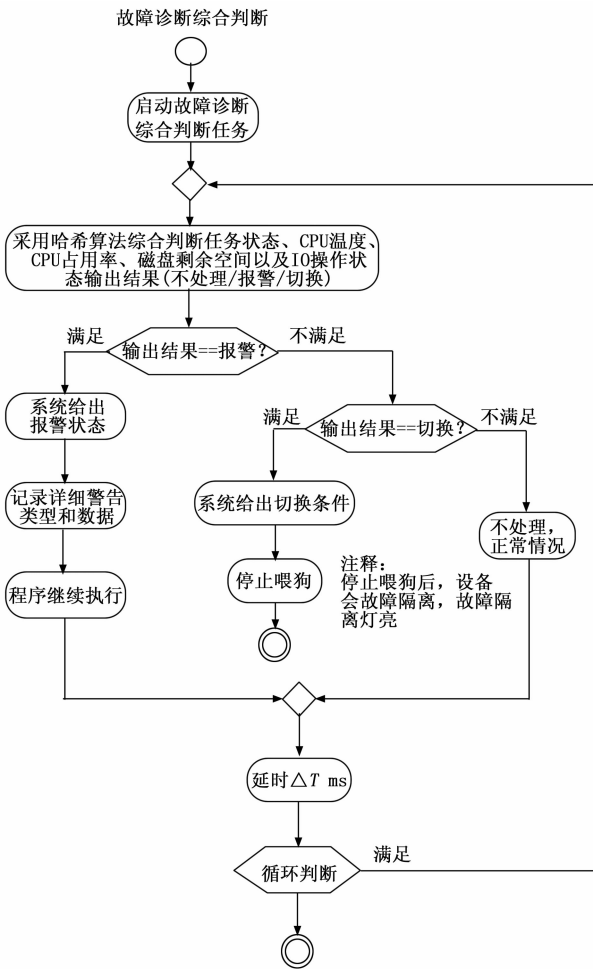


图 14 综合判断逻辑设计

5 实验结果与分析

5.1 本系统实现触发设备切换的技术参数

- 1) 任务状态出现“STOP”;
- 2) CPU 温度 $T_{CPU} \geq 85\text{ }^{\circ}\text{C}$;
- 3) CPU 占用率 $R_{CPU} \geq 80\%$;
- 4) 磁盘剩余空间 $V_{HDC} \leq 1\text{ GB}$;
- 5) IO 操作异常 $IO_{out} \cong IO_{in}$ 。

5.2 实验步骤和方法

测控软件停止给 RC 模块发喂狗操作, 设备即故障隔离, 触发停止喂狗操作的输入有任务异常、CPU 温度异常、CPU 占用率异常、磁盘剩余空间不足、IO 操作异常等。

5.2.1 任务异常识别仿真

任务异常是指所监测任务状态在运行过程中出现“STOP”状态(任务异常无法工作)。所监测的具有代表性的任务主要包含任务如表 1 所示, 仿真过程中, 通过故障注入软件^[23]向前端测控软件中注入触发任务 STOP 指令, 当前端测控软件响应这一条指令时, 任务在 t_1 时刻状态切

换为“STOP”状态, 故障诊断模块识别到异常状态执行停止喂狗操作, 当前设备故障隔离, 对机切换为单主机工作。

5.2.2 CPU 温度异常识别仿真

CPU 温度表示当前主控制器 CPU 实时温度, 仿真时, 通过故障注入软件向前端测控软件中注入修改 CPU 温度全局变量指令:

- 1) 设置值低于预警值 ($T_{CPU} < 70\text{ }^{\circ}\text{C}$), 无处理;
- 2) 设置值高于预警值小于切换阈值 ($70\text{ }^{\circ}\text{C} \leq T_{CPU} < 85\text{ }^{\circ}\text{C}$), 设置报警标志并写日志记录 CPU 温度预警;
- 3) 设置值在预警阈值上下波动, 模拟虚警的工况, 根据设计通过曲线拟合可以判别出此种虚警情况, 记录虚预警记录即可;
- 4) 设置值达到切换阈值 ($T_{CPU} \geq 85\text{ }^{\circ}\text{C}$), 执行停止喂狗操作, 当前设备故障隔离, 对机切换为单主机工作。

5.2.3 CPU 占用率异常识别仿真验证

CPU 占用率表示当前系统 CPU 内存使用情况, 仿真时, 通过故障注入软件向前端测控软件中注入修改 CPU 占用率指令, 指令执行通过 CPU 设置接口控制当前系统 CPU 占用率:

- 1) 设置值低于预警值 ($R_{CPU} < 50\%$), 无处理;
- 2) 设置值高于预警值小于切换值 ($50\% \leq R_{CPU} < 80\%$), 设置报警标志并写日志记录 CPU 占用率预警记录;
- 3) 设置值在报警阈值上下波动, 模拟虚警的工况, 根据设计通过曲线拟合可以判别出此种虚警情况, 记录虚预警记录即可;
- 4) 如果预测值达到阈值 ($R_{CPU} \geq 80\%$), 按照切换处理; 设置值达到切换阈值 ($R_{CPU} \geq 80\%$), 执行停止喂狗操作, 当前设备故障隔离, 对机切换为单主机工作。

5.2.4 磁盘剩余空间异常识别仿真验证

嵌入系统中磁盘空间有限, 磁盘资源宝贵, 剩余空间必须监测。仿真时, 通过 ftp 工具向设备硬盘写入文件, 消耗硬盘空间:

- 1) 剩余空间值高于预警值 ($V_{HDC} > 3\text{ GB}$), 无处理;
- 2) 剩余空间值高于预警值小于切换值 ($1\text{ GB} < V_{HDC} \leq 3\text{ GB}$), 设置报警标志并写日志记录剩余空间不足预警;
- 3) 剩余空间值在预警阈值上下波动, 模拟虚警的工况, 根据设计通过曲线拟合可以判别出此种虚警情况, 记录虚预警记录即可;
- 4) 如果预测值低于阈值 ($V_{HDC} \leq 1\text{ GB}$), 按照切换处理, 执行喂狗操作, 当前设备故障隔离, 对机切换为单主机工作。

5.2.5 IO 操作异常识别仿真验证

IO 操作 ($IO_{control}$) 异常表示 IO 输出结果 IO_{out} 与回采结果 IO_{in} 不一致, 表示 IO 功能失效, 直接影响系统的执行结果。仿真时, 通过故障注入软件向测控软件中注入修改 IO 回采结果 IO_{in} , $IO_{out} = IO_{in}$ 表示正常状态无需处理; $IO_{out} \cong IO_{in}$ 表示异常, 执行停止喂狗操作, 当前设备故障隔离, 对机切换为单主机工作。

5.2.6 故障诊断综合判断仿真验证

综合设计测试用例组合如表 6 所示。

表 6 实验组合列表

序号	实验组合	备注
1	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	
2	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	
3	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意两种异常)	
4	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	
5	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	
6	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种报警)	
7	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种报警)	
8	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种报警)	
9	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 全报警)	
10	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	
11	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	
12	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意两种异常)	
13	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	
14	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	
15	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种报警)	
16	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种报警)	
17	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种报警)	
18	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 全报警)	
19	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	
20	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	
21	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种异常)	
22	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	
23	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	

5.3 实验数据

依据表设计, 每种条件开展 20 次测试, 测试结果如表 7 所示。

表 7 实验结果列表

序号	实验组合	结果
1	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	无动作
2	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	切换
3	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意两种异常)	切换
4	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	切换
5	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	切换
6	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种报警)	报警
7	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种报警)	报警
8	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种报警)	报警
9	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 全报警)	报警
10	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	切换
11	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	切换
12	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意两种异常)	切换
13	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	切换
14	任务异常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	切换
15	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种报警)	报警
16	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种报警)	报警
17	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种报警)	报警
18	任务正常+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 全报警)	报警
19	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均正常)	报警
20	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意一种异常)	报警
21	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意二种异常)	报警
22	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 任意三种异常)	报警
23	任务报警+(T_{CPU} R_{CPU} V_{HDC} $IO_{control}$ 均异常)	报警

5.4 实验结果分析

通过实验结果表明, 文章所设计的系统故障诊断与容错方法能够 100% 识别所监测任务状态、CPU 温度、CPU 占用率、磁盘剩余空间以及 IO 操作异常的“报警门限”和“切换门限”, 并且能够有效地处理虚报警和虚切换的工况, 有效地增强了冗余测控系统的运行可靠性, 文章目前只考虑了任务状态、CPU 温度、CPU 占用率、磁盘剩余空间以及 IO 状态等 5 项异常信息输入且故障信息分析处理算法比较单一, 与实际工况存在着一些差异, 实际场景中工作参数会更多, 如各个功能板卡工作情况、系统数据来源是否存在严重异常, 并且在数据分析方面可以引入智能化算法分析研究等。

6 结束语

文章提出一种故障诊断与容错方法, 能够识别系统中任务异常、CPU 温度过高、CPU 占用率过大、磁盘剩余空间不足以及 IO 操作异常等故障状态, 并对各项结果进行综合分析判断以及结果等级分类与处理, 有效提高了冗余测控系统工作的可靠性和系统稳定性, 但文章也存在输入参数不足以及数据处理不够智能的问题, 还需要对数据预测和学习进行研究, 将来可以延伸应用到小型化多冗余分布式的测控系统。

参考文献:

[1] 王余伟, 曹 东, 施书成. 实时操作系统 CPU 使用率监测的软件容错研究 [J]. 计算机工程与科学, 2018, 40 (8): 1337-1343.

[2] 解梦迪, 贾 凡, 龚 飞, 等. 一种采用硬件内存搬移同步的热备冗余系统 [P]. 中国: CN108092753A, 2018-05-29.

[3] 解梦迪, 毛鹏飞, 董 磊, 等. 一种任务监测方法及装置 [P]. 中国: CN114265741A, 2022-04-01.

[4] 王亚敏, 吴传开, 李丹鹤. 船舶复杂局域网跨操作系统的监测方法 [J]. 舰船科学技术, 2018, 40 (20): 25-27.

[5] 田 垆, 刘宗田. 最小二乘法分段直线拟合 [J]. 计算机科学, 2012, 39 (S1): 482-484.

[6] 王 飞, 张 明, 晏学义. 基于位图技术创建 HBASE 二级索引的实践 [J]. 信息通信技术, 2021, 15 (3): 78-82.

[7] 袁沐春, 郭育辰. 数据包位图索引压缩算法研究 [J]. 网络安全技术与应用, 2016 (8): 36-37.

[8] 邹继军, 饶运涛. 基于散列表的单片机快速查找算法 [J]. 单片机与嵌入式系统应用, 2003 (11): 7-9.

[9] 陈景亮, 张金石. 计算机软件故障的快速修复 [J]. 计算机安全, 2012 (2): 19-22.

[10] 刘 燕, 周国荣, 徐丽莎. 基于嵌入式工程机械监控和故障诊断系统的设计 [J]. 工业控制计算机, 2008 (8): 34-36.

[11] 钟金凤, 吴敬玉, 张云霞, 等. 基于嵌入式 VxWorks 的卫星姿轨控系统的故障模拟系统及方法 [P]. 中国: CN201610885001.8, 2016-10-10.

(下转第 20 页)