文章编号 1004-924X(2000)06-0582-04

计算机控制系统的可靠性分析

李洁
(中国科学院长春光学精密机械与物理研究所, 吉林 长春 130021)

摘要: 可靠性与技术性能是计算机控制系统的两个最重要的方面,但在具体设计工作中,往往特别强调系统的技术性能而忽略了可靠性,尤其是忽略软件可靠性,这常常导致系统不能正常工作,造成经济损失,甚至危及生命安全。本文介绍了计算机控制系统的可靠性的定量表示方法,分析了软件可靠性及硬件可靠性对系统可靠性的影响,对解决办法作了一些探讨。

关 键 词: 软件可靠性; 硬件可靠性; 干扰; 屏蔽; 冗余
中图分类号: TP311.5 文献标识码: A

1 概 述

1.1 系统可靠性的定量表示

微机控制系统的可靠性通常是指微型机系统在规定条件下,在规定的时间内完成规定功能的能力。可靠性只是个定性的概念。实际工作中,往往需要以量的形式具体表示可靠性的高低,如可靠度、维护率、失效率、平均故障间隔时间(MTBF)、平均维护时间(MTTR)、有效度等。具体解释如下:

假定系统投入运行后,工作了一段时间 t_1 后出现了故障,不得不停机维修。经过一段时间 T_1 的维修后,故障排除,系统又正常运行。这样,在时间坐标轴上, $t_1, t_2, \cdots t_n$ 是系统正常工作时间, T_1, $T_2, \cdots T_n$ 是维护时间,则有:

a) 故障率 λ(失效率)

$$\lambda = \frac{\text{失效次数}}{\text{总工作时间}} = \frac{n}{\sum_{i=1}^{n} t_i}$$

故障率表示单位工作时间内发生故障的次数。

b) 维护率 μ

$$\mu = \frac{\text{维护次数}}{\text{总维护时间}} = \frac{n}{\sum_{i=1}^{n} T_i}$$

维护率表示单位时间内修复的次数。

c) 平均故障间隔时间 MTBF(Mean Time Between Failures)

$$\text{MTBF} = \frac{\text{总工作时间}}{\text{失效次数}} = \frac{\sum_{i=1}^{n} t_i}{n} = \frac{1}{\lambda}$$

它表示系统发生多次故障的情况下,平均连续工作时间。

d) 平均维护时间 MTTR(Mean Time To Repair)

$$\text{MTTR} = \frac{\text{总维护时间}}{\text{维护次数}} = \frac{\sum_{i=1}^{n} T_i}{n} = \frac{1}{\mu}$$

它表示系统进行多次维护后的平均维护时间,即平均故障时间。如果该值很小,表示系统可维护性好,容易修复。

e) 有效度 A(Availability ratio)

$$A = \frac{\text{可工作时间}}{\text{可工作时间} + \text{不能工作时间}} = \frac{\text{MTBF}}{\text{MTBF} + \text{MTTR}} = \frac{\mu}{\lambda + \mu} = \frac{1}{1 + \lambda/\mu}$$

有效度表明在某一特定的瞬间,维持其正常工作的概率。其中 λ/μ 是系统的重要性指标。λ/μ 较大,表明系统不能可靠地工作,运行不久即出现故障,有效度降低。

f) 可靠度 R(Reliability ratio)

$$R(n) = P\{n \text{ 次运行不发生故障}\}$$

收稿日期: 2000-06-10; 修订日期: 2000-10-23

可靠度 R 表明运行 n 次不发生故障的概率。也可表示为

$$R = 1 - \lim_{n\to\infty} \frac{n_f}{n}$$

其中, n 是运行次数, n_f 是 n 次运行中发生故障的次数。

如果按限定的时间计算, 可靠度为

$R(t) = P\{$在时间$[0, t]$内运行不发生故障$\}$

它表明在限定的时间$[0, t]$内发生故障的概率。

1.2 如何提高可靠性

由上可知, 提高可靠性有两个方面: 一是尽量使系统在规定时间内少发生故障和错误; 二是发生了故障能迅速排除。为了提高微机控制系统的可靠性, 通常可从硬件可靠性及软件可靠性两方面来解决。硬件主要考虑如何提高元器件和设备的可靠性; 采用抗干扰措施, 提高系统对环境的适应能力和冗余结构设计。软件主要考虑测试技术、故障自诊断技术、自动检错、纠错技术、系统恢复技术方面的设计。

2 硬件可靠性

控制系统软件的可靠性与硬件的可靠性是密切相关的, 在此简要说明一下硬件的可靠性问题。研制微机控制系统时, 首先要根据其性能指标和功能要求决定系统的结构形式, 确定软硬件的分工和电路具体结构。在系统的硬件方案设计时应考虑的问题:

a) 系统的可靠性是由系统中的各个元件的可靠性决定的。只要能满足系统的性能指标, 就应尽可能地简化系统结构, 减少元件的数量, 简化方案。

b) 避免片面追求高性能指标和过多的功能。

c) 合理划分软硬件功能, 在 CPU 时间资源允许的前提下能够方便地用软件完成的功能一定要用软件实现。

d) 许多元器件的失效与温度有密切的关系, 热设计的正确与否是影响系统工作稳定性及可靠性的主要因素之一。

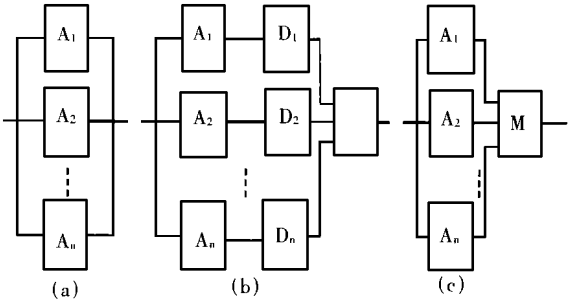
e) 元器件的引脚焊点、模块间的接插件、总线插件等的电器互连是电子电路中故障率较高的部分, 需要高度重视。

f) 振动会对系统的元器件及整机结构造成机械性损坏, 因此必须有机械防震设计, 其重点在于系统接插件、系统模块和质量较大器件的固定。

g) 根据系统可能工作的环境进行防护设计, 通常需要考虑温度、湿度、气压、雨、雪、盐雾、腐蚀性气体、沙尘及辐射等。

h) 抗干扰技术中解决干扰问题, 历来是硬件设计中一个伤脑筋的问题。干扰源、传输途径及干扰对象是构成干扰的三个要素。常用的抑制干扰的措施主要有: 滤波、接地、屏蔽、隔离、设置干扰吸收网络及合理布线等。采用合理的接地技术、屏蔽技术、隔离技术、以及长线传输、电源干扰的抑制等手段提高系统的抗干扰能力。

i) 冗余技术。故障容错是利用冗余的元部件来屏蔽已发生的故障对系统的影响。常用的冗余系统, 按其结构可分为并联系统、备用系统和表决系统三种, 如图所示。



(a) Parallel connection system　(b) Reserve system　(c) Decide by system

Fig. 1　Structure of the redundancy system

3 软件可靠性

3.1 利用软件提高系统的可靠性

由于系统是由硬件和软件组成的, 因而系统的可靠性也分硬件可靠性和软件可靠性两个方面。通过提高元器件的质量、采用冗余设计、进行预防性维护、增设抗干扰装置等措施, 能够提高硬件的可靠性, 但是要想得到理想的可靠度是不够的, 通常还要利用软件来进一步提高系统的可靠性。具体措施包括:

a) 采用系统信息管理的软件, 用软件进行系统调度。当发生故障时进行现场保护, 迅速用装置代替故障装置; 在过负荷时采取应急措施; 在故障排除后使系统迅速恢复正常运行。

b) 编制诊断程序, 及时发现故障并排除。这里主要介绍两种措施。

(1) 程序运行监视系统

程序运行监视系统 WTD, 直译为 "看门狗", 是一种软硬件结合的抗程序 "跑飞" 的措施。其硬

件是一个用于产生定时 T 的计数器或单稳触发器。其定时输出端接至 CPU 的复位线, 而其定时清零则由 CPU 控制。在正常情况下, 程序启动 WTD 后, 即以小于 T 的间隔 t 将其清零一次, 这样 WTD 的定时溢出就不会发生。在受到干扰的异常情况下, 程序的正常执行顺序被破坏, 不可能周期性地将 WTD 清零, 导致 WTD 定时溢出, 使系统复位, CPU 摆脱因程序 "跑飞"造成的瘫痪状态。WTD 是一种被动的抗干扰措施, 它只能在一定程度上减少干扰造成的损失。其困难之处在于时间T 长短的选择较困难。

(2) 软件陷阱

软件陷阱是指令冗余的一种应用形式, 用于捕捉 "跑飞"的指令指针。对于因受干扰而混乱的程序, 多字节指令是最危险的, 因为错误的 PC 指针很有可能落在多字节指令的中间, 造成指令错误执行; 而单字节指令则可使混乱的 PC 指针重新理顺, 使混乱现象得以控制。软件陷阱就是根据这个道理构成的一个程序段。以下是与 MCS- 51 单片机相对应的一个软件陷阱:

NOP 00

NOP 00

LJMP ERROR 02 XXXX

上诉程序中 LJMP ERROR 将 "跑飞"的程序指针转移到出错处处理程序。其 NOP 指令加得越多, 捕捉能力越强。软件陷阱通常安排在下列位置:

. 系统中空的 ROM 区。

. 数据表格的头尾处。

. 程序中未用的中断向量处。

. 程序内如跳转指令等语句之后。

软件陷阱对陷入死循环的 "跑飞"程序无能为力, WTD 这方面则更可靠, 可以将两者结合使用。

c) 指令复执技术

指令复执技术是指在程序执行过程中, 一旦发现错误就重新执行被错误干扰的现行指令。指令复执既可以用软件方法实现, 也可以用硬件方法实现。其实现必须遵循下面两点:

(1) 发现错误时, 应能准确地保留现行指令的地址, 以便重新执行。

(2) 应能保留现行指令所用的数据, 以便在重新执行时使用。

指令复执的次数可用次数控制及时间控制两种方法。

d) 输入输出软件抗干扰技术

(1) 对模拟量多次采样。

(2) 为确保开关量正确输入, 可采取多次读入并比较的方法。

(3) 重复输出同一数据是软件最为有效的输出抗干扰措施。

3.2 提高软件自身的可靠性

软件可靠性是在规定的时间内和规定的环境下, 计算机程序无故障运行的概率。"软件故障"是指程序运行的外部结果偏离了需求规范。所以 "故障"是在动态中产生的, 必须执行程序才会发现故障, 故障与程序运行状态有关。

提高软件自身的可靠性包括两个方面: 一是采取措施, 减少软件设计中的错误, 这包括采用模块化设计、进行软件评审和对软件进行测试等; 二是采用提高可测试性的设计, 在作系统设计时就充分考虑到测试的要求, 使得软件的可维护性较高、故障的诊断及时迅速。

3.3 测试技术

大量统计资料表明, 软件测试的工作量往往占软件开发总工作量的 40% 以上, 在极端情况, 测试关系人的生命安全的软件所花费的成本, 可能相当于软件工程其他步骤中成本的三到五倍。因此, 必须高度重视软件测试工作。测试阶段的根本目标是消除故障保证软件的可靠性。软件故障占整个系统故障的 65%, 因此保证软件的可靠性就更重要了。

测试是质量控制的措施之一。测试就是软件的动态执行过程以及执行结果与已知的预期结果相比较。测试分为三类: 一般测试、特殊测试和包含用户的测试。

一般测试涉及各类软件, 目的是试图去除诸如分支错误、循环错误、不正确输出等共同类型的缺陷。特殊测试, 常常针对整体安装中出现的问题或性能降低问题。包含用户的测试, 主要针对可使用性问题以及确保所有需求实际实现。

上述测试又可划分为白盒测试, 黑盒测试及混合测试。其中白盒测试包括: 子程序测试, 单元测试, 病毒预防测试, 能力测试, 性能测试, 安全性测试, 2000 年问题测试。黑盒测试包括: 全面应用的系统测试, 新功能测试, 实验室测试, 可使用性测试, 客户验收测试, 现场测试, 净室统计测试。混合测试包括: 独立测试, 回归测试, 组装测试, 平台测试。

3.4 故障自诊断技术

早期的数字系统故障诊断是依靠工程技术人员凭借自己的丰富经验和理论知识, 并借助一些常规的仪器来完成的, 速度较慢, 效率较低。随着计算机系统规模的扩大和结构的日趋复杂, 人工诊断愈来愈困难, 而计算机技术的迅速发展, 使故障检测和诊断的自动化成为可能。目前, 输出故障诊断的基本方法是: 向被诊断的部件或装置输入一串数据(称为测试码) , 观测相应的输出数据(称为校验码) 。根据事先已知的测试码、校验码和故障的对应关系, 通过对观测结果的分析以确定是否发生故障, 以及故障发生的部位。

为了使故障诊断具有良好的性能(时间短, 准确度高, 分辨能力强) , 选择测试码、确定校验码和故障之间的对应关系、研究实际可行的诊断方法是故障诊断的关键。同时人们发现, 即使有比较成熟的理论指导, 对许多复杂系统故障的诊断所需的时间 和工作量也是巨大的。因此, 需要在系统设计的时候, 考虑到故障诊断的要求, 使系统易于测试, 既要在故障诊断理论的指导下设计系统, 称为可测试性设计。这是故障诊断技术的两个主要的发展方面。

4　结　束　语

值得指出的是, 人们越来越注重系统的可靠性设计, 但相对而言, 对软件工程和软件可靠性工程的认识和重视还很不够, 既有认识上的问题, 也有技术、经费上的问题, 在软件开发研制进程中, 管理方面亦存在较大的问题, 这些都应该引起人们高度的重视。本文讨论的关于提高软、硬件可靠性的方法还很有限, 例如: 软件可靠性增长建模、串模干扰抑制、共模干扰抑制等, 有待今后进一步探讨。

参考文献:

[1] 刘明俊, 杨壮志, 张拥军, 郭鸿武. 计算机控制原理与技术[M]. 长沙: 国防科技大学出版社, 1999.

[2] 蔡开元. 软件可靠性工程基础[M]. 北京: 清华大学出版社, 1995.

[3] ROBERTL. GLASS. Software Reliability Guidebook[M]. New Jersey, Ptentice- Hall, Inc. Englewood, 1979.

[4] 穆沙 J D, 艾里诺 A, 奥本 K. 软件可靠性- 度量、预计和应用[M]. 姚 一平, 林典伦, 裘忠侯, 才 全译. 北京: 机械工业出版社, 1992.

[5] 石剑琛, 贵可荣, 汤志国. 测试方法综述[J]. 武汉大学学报, 1999, 45(5): 687- 690.

Reliability analysis of computer controlled systems

LI Jie

(*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun* 130021, *China*)

Abstract: Reliability and function of the technology are two important aspects of the computer controlled systems. But in the concrete design, people often concentrate more on the function of the technology than the reliability, especially the software reliability. This ignorance leads to the system out of control and results in the economic loss, even endanger people`s life. This article introduces the computer controlled system reliability in quantity. The effects of software reliability and hardware reliability on the systems reliability are analyzed. The methods of solving the problems are discussed.

Key words: software reliability; hardware reliability; interference; shield; redundancy

作者简介: 李洁(1973-), 女, 吉林市人。中国科学院长春光学精密机械与物理研究所在读硕士研究生, 主要从事计算机应用的研究。