

基于 Python 语言的余度特性分析与应用

邱 菊, 叶志锋, 赵永平

(南京航空航天大学图书馆, 江苏 南京 211106)

[摘要] 针对余度飞控计算机系统各方面指标要求,介绍了基于 Python 开发语言的余度飞控计算机系统. 提出了一种新的理念——预处理理念,在这种理念的基础上,并建立了硬件三余度和软件三余度的架构模型. 其中,在软件三余度架构模型中,3 个软件版本由不同语言开发,第一个软件和第二个软件用 Python 语言开发,第三个软件用 C 语言开发,在三台计算机上互为主备份软件. 结果表明,该方法在提高可靠性故障维修率的同时可以降低开发成本及同态故障率.

[关键词] Python, 余度, 飞控系统, 可靠性

[中图分类号] V19 **[文献标志码]** A **[文章编号]** 1672-1292(2018)04-0080-06

Analysis and Application of Redundancy Characteristics Based on Python Language

Qiu Ju, Ye Zhifeng, Zhao Yongping

(Library, Nanjing University of Aeronautics & Astronautics, Nanjing 211106, China)

Abstract: In view of the requirements of all aspects of the redundancy flight control computer system, the redundancy flight control computer system based on Python development language is introduced. A new concept, pretreatment concept is put forward on the basis of this idea, the structure models of the hardware three redundancy and software three redundancy are established. Among them, in the software three-redundancy architecture model, three versions of software are developed in different languages. The first software and second software are developed by Python, the third software is developed by C language, each other being the backup software on the three computers. The results show that the method can reduce the cost of development and the homomorphic failure rate while improving the reliability and failure maintenance rate.

Key words: Python, redundancy, fly control system, reliability

随着航空领域计算机及通信技术的不断发展,飞行控制系统在过去的几十年中发生了翻天覆地的变化,电子计算机、传感器、电力驱动等装置逐渐替代原始的机械控制系统. 在这些系统中,实时软件可以优化被控制系统的性能和效率. 当电子元器件出现故障时,为了确保这些系统的可靠性,通常要对计算机、传感器、软件以及其他电子元件等进行余度配置^[1]. 我国飞机控制系统较发达国家有很大程度的落后,为了提高可靠性,必然要增加余度. 但一味地增加余度,又会增加开发成本和飞机总重量. 除了满足安全性的要求,飞机还得满足重量要求. 因为飞机工作环境的特殊性,相对于其他地面工程设计,飞机结构重量要求有其特殊的重要意义. 在总重量保持不变的情况下,如果在设计的过程中增加飞机重量的比重,则意味着减少有效载重,或降低飞机性能(比如航程,因为储油量会变少),而有效载重和飞机性能是飞机战斗力或运载效能的主要因素^[2]. 所以,要在满足飞机结构寿命的前提下,保证结构重量最轻. 在飞机构造发展历程中,这一要求是推动飞机结构型式、航空材料等发展的重要因素之一. 在飞机设计中,减轻结构重量不是一件很容易办到的事情. 飞机由成千上万个零构件组成,如果设计或制造不当,每一个零构件稍微超重,积少成多就会使整机超重很多,这将导致设计失败或产品报废. 因此在飞机整个研制过程中,都要进行严格的重量控制. 总之,较其他工程构造设计而言,更要强调“精心设计、精心施工”^[3].

收稿日期:2018-04-18.

通讯联系人:叶志锋,博士,教授,研究方向:航空宇航推进理论与工程. E-mail: yzf@nuaa.edu.cn

基于上述安全和减轻重量的综合考虑,特提出基于 Python 开发语言的余度飞控系统. 往常开发语言基本都选用 C 语言,这样做的缺点是版本之间互异性很小,发生同态故障的概率很大,另外,在数据封装性方面,C 语言有很大的缺陷,而且 C 语言对变量的类型约束不严格,这也就导致了开发出来的程序在数据的安全性上有很大缺陷. 而 Python 语言是一种集简洁性、易维护性、速度快与一体的程序设计语言,提出用 Python 语言作为软件模块开发的首要语言,这可以增加软件的安全性,适当减少余度数量,减少开发成本. 但是为了减少同态故障发生的概率,不能所有的软件版本都用一种语言开发,所以安排一个版本的软件用 C 语言开发. 软件模块方面,后面会进行数学分析,用三余度是最合适的,在保证安全性能的同时还能尽可能减少不必要的开支. 硬件模块余度数量方面,经过数学分析,用三余度或四余度是最合适的,但四余度会增加一些开支和飞机的重量,在经过思考和研究之后,决定在硬件方面实行三余度,同时,在错误监测隔离修复模块(硬件的备份余度与这个模块直接相连)前面,添加一个模块,即预处理模块,这个模块,能把 90% 的错误通过重启解决,剩下一些不能通过重启来解决的错误,通过错误监测隔离修复模块解决. 这样,在提高软硬件安全性的同时,又没有过分增加余度,飞机重量控制在合理的范围内,经过实验,本方法具有很好的应用价值^[4].

1 预处理模块

飞机在空中飞行过程中,出现的错误,都是通过错误监测隔离修复功能模块实现^[5]. 这个模块决定要不要调用余度单元切换到其他分支以继续正常运作. 但在使用过程中会发现一个问题,其实大多数故障或错误都是假的、瞬间的或者是很小的错误,一般情况下这种错误通过简单的重启操作就能修复. 若每次一出现故障或错误就立马启动错误监测隔离修复功能模块,调动余度模块,让备份模块开机进行冷启动,会造成很大的成本. 对于一个永久的单元模块错误来说,调动余度单元切换到其他分支是非常正确的,但是这种概率很小. 一般情况下的错误都很简单,一个小小的重启操作就能解决问题,进而让程序模块恢复正常. 所以,在本设计方案中,在错误监测隔离修复功能模块前加一个预处理模块,这个模块在错误监测隔离修复功能模块之前使用,每次出现错误时,先启动该模块,大多数的错误都可以通过这个预处理模块进行修复,对于那些不能通过预处理模块进行修复的恒久错误,就通过错误监测隔离修复功能模块启动余度单元,切换到其他分支. 重启操作简单并且迅速,并且对于一个错误模块来说,重启操作只是尝试去解决问题,并不会给系统和硬件带来更进一步的破坏,同时也不会影响到一个正常的功能模块. 当设计预处理模块时,关键点就是通过增加这个预处理模块,让系统模块与错误监测隔离修复模块之间的交互最小. 从而尽量减少切换到备份模块的次数,这可以在保证飞控系统安全性的前提下尽量缩减软、硬件余度数量. 本设计方案如图 1 所示.



图 1 添加预处理模块的程序模块结构图

Fig. 1 The program module structure diagram of add preprocessing module

2 硬件余度方案设计

余度,即在只有一个单独故障时能正常工作,当出现一个以上的独立故障时,才能引起既定的不希望工作状态的一种设计方法.

余度设计的基本任务包括:容错能力准则、部件余度类型、系统余度配置等方面. 其中,容错能力准则是以满足任务可靠性为目标. 但是,在提高可靠性的同时,余度配置也带来了一些问题. 系统的造价成本增加,同时设备的重量、体积和线缆数量、开发费用、维修费用也有增加. 所以如何在保证飞行控制系统安全系数的同时尽可能缩小成本,即控制余度数量,成为了飞控系统方面研究的一个重点.

2.1 冗余结构模式

随着提高硬件可靠性需求的提出,便有了冗余技术这个概念. 冗余,即多余的资源. 在系统由于一个分支出现问题而出现故障时,就隔离掉故障的部分,由冗余的部分替代^[6]. 它能提高任务可靠性和系统安全性^[7].

冗余的结构,有以下几种模式:(1)主动冗余,也叫做“热备份”冗余,是最常用的模式.在这种工作模式中,在待命状态下,冗余模块和工作模块一样施加所有的系统工作信号,并供给全额电源.(2)“冷备份”冗余,与第一种相反,在这种工作模式中,在待命状态下,冗余模块完全不施加任何系统信号,在电源方面,完全没有供给.(3)“温备份”冗余,介于(1)和(2)之间,在这种工作模式中,在待命状态下,仅仅施加部分系统工作信号,在电源方面,也是供给一部分电源.(4)是最复杂的一种冗余结构.在这种工作模式中,一部分冗余模块是主动结构,还有一部分冗余模块是被动结构.在一般飞行控制系统中,多采用主动冗余结构.

2.2 总线的由来

20世纪60年代以前,航空电子系统是非常简单、独立的一种系统,航空、通信、飞行控制和显示器等均由模拟系统构成.信号的主要构成是模拟电压、同-异步信号和接触式开关.随着数字技术的出现,航空设备系统中也逐渐引入数字计算机,随着数字技术的不断发展和完善,航空电子设备系统也跟着变得越来越数字化.为了应付不同的航空设备,航空电子设备之间的通信需要不同的硬件接口来解决,这直接导致通信十分混乱,安全性也降到很低.为了方便航空电子设备及接口之间的通信,提出数据总线的概念,即在不同的航空电子设备和不同的时刻之间能相互通信^[8].

总线就是通过多路传输,把机上各机载电子设备连接到一起,组成一个网络,为航空电子系统提供综合化、集中式的系统控制和标准化接口.实现控制信息的正确传输和共享,比较具有代表性的有1553B总线和ARINC429总线,本文采用1553B总线^[9].为了提高子系统和全系统的可靠性,要进行冗余度设计.

2.3 数学公式分析

P 为飞控计算机的失效概率, p 为每个总线通道的失效概率, a 为故障测试的覆盖率, m 为冗余通道数.在冗余模型中,有一个计算机是主控计算机,用来产生系统输出,其他备份计算机用来作为冗余备份.当主控计算机失效时,立即把备用计算机切换到工作状态,进而保证系统的正常运行.当最后一个通道失效时,冗余系统就宣告失效.现定义,当 $m=3$ 时,有

$$P=(p(1-a)+p^2a(1-a)+p^3a^2(1-a))\cdot 10\%. \quad (1)$$

系统的失效概率是 a 的函数.在实际操作中,如果 $a=0.99$,假设有4个通道,那么模型的失效概率是 $10^{-6}/h$,不能保证系统有足够的可靠性,所以要使用比较监控技术.

假设通道数为 m ,运行产生的结果送到表决器,如果通道数目 $m>2$,那么 $a=1$.当只有2个通道的数据被送至表决器参加表决时,表决无法进行,此时就得通过机内测试检测故障,因此有:

$$P=p^m+3p^{m-1}r(1-a). \quad (2)$$

当通道数为1时,不能满足要求,当通道数为2时,也只能勉强满足系统要求.如果再加一个通道,把通道数增至3时,若 $a=0.99$,系统的失效概率级别为 $10^{-12}/h$,可以满足系统可靠性的要求.

2.4 总线冗余设计

经过上述分析,结合预处理模块思路,得出结论,对总线实行三冗余方案设计是最合理的,图2是总线冗余设计方案.

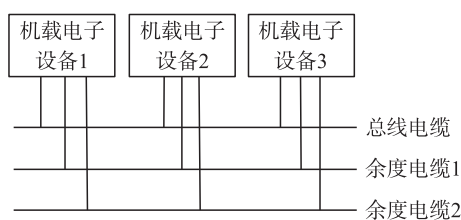


图2 总线冗余设计方案

Fig. 2 Bus redundancy design scheme

3 软件冗余方案设计

3.1 软件可靠性

软件的可靠性不能通过冗余实现,为了提高软件可靠性^[10],需要适当增加处理器上软件版本的数量,当一个版本的软件出现故障时,备用的软件版本变成当前运行的软件版本.但一味地增加软件版本,研发成本和硬件成本势必会增加,所以应在保证安全系数的同时尽可能减小成本^[7].除此之外,还应该尽量保持软件可靠性和硬件可靠性的均衡,否则其中一方会成为系统可靠性的瓶颈.而且,如果软件版本过多,版本间的差异性会减小,那么发生同态故障的概率就会增加.因此,经过综合考虑,三冗余软件架构是最佳选择.

3.2 开发语言的选取

通常习惯性选择 C 语言作为程序设计开发的优选语言,因为 C 语言易学通用. 但是当大量的开发人员相互之间需要更新大段代码时,C 语言可能会使事情变得很麻烦. 并且,在输入输出方面比较复杂^[11-13]. 而相对 C 语言来说,Python 语言有很多 C 语言没有的优点. Python 语言是一种面向对象的解释型计算机程序设计语言,是纯粹的自由软件,语法简洁清晰. 另外,Python 具有丰富和强大的数据库,它常被称为胶水语言,能够把其他语言制作的各种模块(尤其是 C/C++)很轻松地联结在一起. 在这方面,如果开发团队里面不同组员使用不同的软件设计语言^[12],可以用 Python 将这些程序设计模块组合到一起,而不用再找人重新开发,方便快捷. Python 里面有很丰富的 API 和工具,可以使用 Python 快速生成程序的模型包括程序的最终界面,然后对其中有特殊要求的部分,用更合适的语言(如 C 语言、C++、Python 等)来进行改写,然后进行集成和封装. 在可维护性方面,因为 Python 语言的简洁易读性和可扩展性,当出现故障时,维护人员可以很方便对程序进行维护. 2004 年 Google 使用 Python 作为胶水语言进行使用,使用 C++编写性能要求极高的部分,接着用 Python 调用相应的模块. 在快速开发时使用 Python,在操控硬件的场合使用 C++.

飞行控制系统对响应时间、系统可靠性及效率、瞬态故障的可恢复性、维修成本等都有很高要求. 基于上述优点,选用 Python 作为本次开发系统的首选程序设计语言,第一个软件版本和第二个软件版本用 Python 进行开发,第三个软件版本用 C 语言开发,分别记为软件 a,软件 b,软件 c^[14].

3.3 数学公式分析

记软件系统的复杂性为 M ,在软件系统上面所做的投入记为 E ,软件系统的设计方法记为 F ,比例因子用 k 来表示. 用可靠性函数 R 来对软件系统的可靠性进行描述,

$$R(M,E,F)=e^{\frac{kMi}{EF}}.$$
 (3)

对软件系统的设计进行可靠性和差异性的均衡,可以参照下面的 2 个准则.

- (1) $R_s>R$,这意味着对于软件系统来说,它的可靠度要比系统的可靠性指标大.
- (2) $\sum_{i=1}^v Ei<E$,式中, Ei 是第 i 版本软件的投入, v 是软件版本的个数, E 是成本预算的上限,所有版本软件的投入的加和,不能超过成本预算的上限.

在多版本程序设计中,想尽可能地提高系统可靠性,必须在不超过成本预算上限的前提下. 另外,如果盲目增加软件版本,会让版本间的差异变小,这会导致同态故障发生的概率大大增加,而且也会增加开发成本. 经过综合考虑,软件版本三余度是最合理的设计方案.

3.4 软件余度方案设计

在本设计方案中,每台计算机上安装有三个版本软件,分别为软件 a、软件 b、软件 c. 在计算机 A 上,软件 a 为主运行软件,软件 b 为一号备份软件,软件 c 为二号备份软件,若软件 a 运行过程中出现故障,启动故障修护功能,软件 b 转为主运行软件. 若软件 b 在运行过程中出现故障,那么启动故障修护功能,软件 c 转为主运行软件. 计算机 B、计算机 C 软件版本的运行算法与计算机 A 相似. 这在降低开发成本保证软件运行可靠性的同时很好地规避了同态故障,图 3 是软件余度方案设计图.

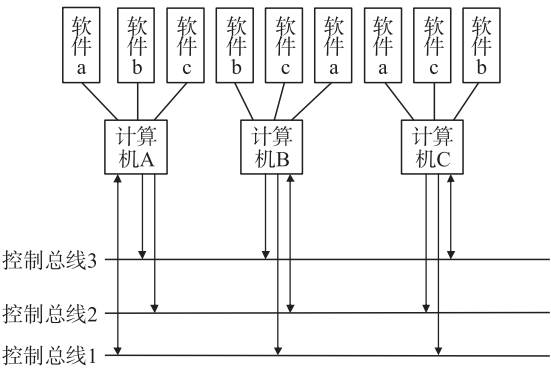


图 3 软件余度方案设计图
Fig. 3 Software redundancy program design drawing

4 飞控计算机余度总设计

4.1 比较监控余度技术

前面提到比较监控余度技术,是指每个通道均使用两台计算机进行比较监控,当这两台计算机的输出差值超过规定的阈值时,切除该通道的输出.

4.2 余度飞控计算机系统架构

在本设计方案中,硬件通道采用三余度方案,有三台计算机、三个通道,启用比较监控余度模型. 每个

通道里面有两个支路,这两个支路互相监测.当这两个支路输出的差值超过规定的阈值时,设为故障,切除该通道的输出,启动预处理模块,进行重启恢复,对于恢复不了的通道,设为故障,不再启用.对于三余度飞控系统,一个通道的计算机作为主控计算机,用以产生系统输出,剩下两个通道的计算机作为备份.当主控计算机出现故障时,按照优先级从剩下两个通道的计算机中挑出一个,切换到工作状态,以保证系统的正常运行.每个计算机中有三个版本的软件,分别为主、备份软件.在当前主运行软件出现故障时,切换到优先级高的备份软件,若优先级高的备份软件出现故障,则切换到第二备份软件.

当A、B、C三个通道的离散量加一起等于3,表示3个通道均工作正常.当A、B、C三个通道的离散量加一起等于2,那么取值为0的那个通道工作异常,启动预处理模块进行工作恢复,其他两个通道工作正常.当A、B、C三个通道的离散量加一起等于1,那么分别对取值为1的通道和取值为0的通道进行比较监控,启动预处理模块进行故障修复,若修复不了,启动错误检测隔离修复模块,调动余度单元,及时对故障通道进行隔离和修复^[15-16].

5 结语

此余度飞控系统,采用了硬件通道三余度、软件版本三余度、比较监控技术的设计方案,并添加了一个预处理模块.在出现故障时,首先启动预处理模块,这个功能模块可以解决90%以上的错误.剩下一些永久性的故障,再启动错误监测隔离修复模块进行调用余度单元,并把错误分支隔离.硬件通道方面,三余度方案能使得可靠性增加.软件方面,与传统的所有版本都是C语言开发方式相比,Python语言开发方式简单清楚、易于维护管理、方便扩展,减少了开发人力和开发时间,并且易于后期维护,另外加上C语言版本软件互为主备份软件.

[参考文献](References)

- [1] 杨开全,叶志锋,万云.基于ARM微控制器的FPGA配置及应用[J].测控技术,2007,26(7):62-65.
YANG K Q, YE Z F, WAN Y. Configuration and application of FPGA based on ARM micro-controlle[J]. Measurement & control technology, 2007, 26(7): 62-65. (in Chinese)
- [2] 施坤娣.飞机构造与强度基础[M].北京:航空工业出版社,1989:5.
SHI K D. Aircraft structure and strength foundation[M]. Beijing: Aviation Industry Press, 1989: 5. (in Chinese)
- [3] 姚一平.可靠性及余度技术[M].北京:航空工业出版社,1995.
YAO Y P. Reliability and redundancy technology[M]. Beijing: Aviation Industry Press, 1995. (in Chinese)
- [4] 杨伟,章卫国,杨朝旭,等.容错飞行控制系统[M].西安:西北工业大学出版社,2006.
YANG W, ZHANG W G, YANG Z X, et al. Fault tolerant flight control system[M]. Xi'an: Northwestern Polytechnical University Press, 2006. (in Chinese)
- [5] 刘小雄,陈怀民,章卫国.自监控二余度飞控计算机系统设计[J].测控技术,2005,24(7):72-75.
LIU X X, CHEN H M, ZHANG W G. Design of self-monitoring dual redundancy flight control computer systems[J]. Measurement & control technology, 2005, 24(7): 72-75. (in Chinese)
- [6] 刘小雄,章卫国,黄宜军.解析余度关键技术研究与发展趋势[J].计算机测量与控制,2005,13(7):710-713.
LIU X X, ZHANG W G, HANG Y J. Research methods and development of analytical redundancy technology[J]. Computer measurement & control, 2005, 13(7): 710-713. (in Chinese)
- [7] 翟小花.余度飞行控制系统低成本方案研究[D].南京:南京航空航天大学,2005.
ZHAI X H. Research on a low cost solution for redundant flight control system[D]. Nanjing: Nanjing University of Aeronautics and Astronautics, 2005. (in Chinese)
- [8] 丁志平.一种基于FPGA的航空总线容错机制设计[J].微型机与应用,2015,34(10):20-24.
DING Z P. A new avionics bus fault tolerance mechanism design based on FPGA[J]. Microcomputer & its applications, 2015, 34(10): 20-24. (in Chinese)
- [9] YANG J X, ZHANG J. Research on test of airborne dual-redundant data bus of 1553B[J]. Computer measurement & control, 2010, 18: 1962-1963.
- [10] EDWARD H P. The electric jet: F-35 uses high-speed data bus; F-35's vehicle network integrates flight controls, propulsion

- and subsystem processing[J]. Aviation week and space technology,2007,166(6):56.
- [11] BAZHENON S G,LYSENKOVA N B. Selection method of monitoring algorithm thresholds for airliner's digital fly-by-wire control system[J]. TsAGI science journal,2015,46(1):63-74.
- [12] XUE Y,YAO Z Q,NIU W. The distributed dissimilar redundancy architecture of fly-by-wire flight control system[C]// Proceedings of 2016 12th International Conference on Computational Intelligence and Security. IEEE,2016.
- [13] HUTTO A J. Flight-test report on the heavy-lift helicopter flight-control system[J]. Journal of the American helicopter society, 1976,21(1):32-40.
- [14] LIU X X,ZHANG W G,LI G W. Redundancy techniques for fly-by-wire flight control systems[J]. Aircraft design,2006, 21(1):35-38.
- [15] GOUPIL P. Oscillatory failure case detection in the A380 electrical flight control system by analytical redundancy[J]. Control engineering practice,2010,18(9):1110-1119.
- [16] YEH Y C. Triple-triple redundant 777 primary flight computer[J]. Aerospace applications conference,1996,1(1):293-307.

[责任编辑:陈 庆]