

面向空空导弹飞控软件的软件容错技术研究

李 伟

(中国空空导弹研究院, 河南 洛阳 471009)

摘 要:随着临近空间飞行器的发展,空空导弹作战空域也向超高空域拓展,因此在空空导弹设计时需要考虑高能粒子引起的计算机系统瞬时故障。本文从软件防护的角度出发,针对该故障提出了一种基于冗余备份及校验和的软件数据流加固方法。试验表明,该算法对单粒子翻转引起的瞬态故障具有良好的防护性,具有通用性好、易移植等优点。

关键词: 飞控软件; 软故障; 单粒子翻转; 软件容错; 数据流错误恢复

中图分类号: TP302.8 **文献标识码:** A

Software Fault-Tolerance Techniques Research for Air-to-Air Missile Control Software

LI Wei

(China Airborne Missile Academy, Luoyang 471009, China)

Abstract: With the development of near-space vehicle, the combat airspace of air-to-air missile is also extended to the ultra-high airspace. Therefore the design of air-to-air missile needs to consider the transient fault of the computer system caused by high-energy particles. From the point of software protection, this paper proposes a software data-flow reinforcement method based on redundant backup and checksum. The experimental results show that the algorithm has the advantages of good protection against transient faults caused by single event upset, and has the advantages of good universality and easy transplantation.

Keywords: control software; software error; single event upset; software fault-tolerance; data-flow error recovery

1 引言(Introduction)

近年来临近空间飞行器的发展大大加快,空空导弹的作战空域也需要从传统的低空域向临近空间超高空域拓展^[1]。而临近空间超高空域的空间辐射环境远比空空导弹传统低空域作战环境复杂,在临近空间中电子、光子、中子及质子等高能粒子的辐射强度大大增加,这些高能粒子辐射强度的增加会对电子设备的正常运转造成严重影响。这种现象称之为高能粒子辐照的单粒子效应(Single Event Effects, SEE),具体又可分为单粒子翻转(Single Event Upset, SEU)、单粒子门锁(Single Event Latchup, SEL)和单粒子烧毁(Single Event Burnout, SEB)。如果空空导弹在该空域作战,需要考虑这些影响。而空空导弹工作周期较短,一般不会发生单粒子门锁或单粒子烧毁,因此需要重点考虑的是单粒子翻转造成电子器件内部的逻辑状态改变,即器件内部的数据状态在0和1之间的转变。这种转变会形成瞬态故障,影响系统的正常

运行。

目前,在单粒子效应防护方面可分为硬件防护和软件防护,硬件防护^[2]方面有器件屏蔽设计、电路加固设计及硬件冗余等技术,软件防护方面有指令冗余^[3]、多线程^[4]、进程冗余^[5]等技术。因此,需要在该领域作战的空空导弹可靠性设计时除了在硬件方面进行必要的屏蔽防护外,在软件方面也需要考虑采用必要的防护措施。

本文从软件的角度对单粒子效应引起的瞬态故障的防护机制进行研究,提出了一种基于冗余备份及校验和的软件数据流加固方法(Backups-Check Software Error Detection and Recovery, BSED-R),可对空空导弹嵌入式飞控软件中的部分核心代码进行加固实现,该方法相对于硬件防护具有灵活性高、通用性好、易于移植及成本低等优点。

2 系统设计(System design)

软件的数据流错误是指单粒子翻转引起的瞬态故障发生

在系统的数据存储或传输单元,在数据的存储或者传输过程中发生了数值跳变,导致程序流程虽然正常运行但运算结果出现错误,这将给系统带来严重后果,特别是某些重要数据如发生错误甚至会造成灾难性后果。这对于空空导弹这样一种对可靠性要求极高的系统来说是不可接受的,因此有必要采取有效地容错机制来检测和纠正这种数据流错误。

一个完整的软件数据流错误容错处理机制如图1所示。

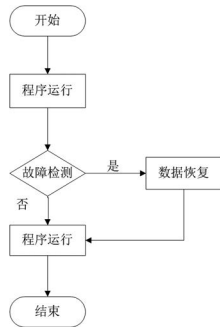


图1 软件数据流容错处理机制流程图

Fig.1 Software data-flow error recovery processing mechanism flow chart

软件数据流错误容错处理机制一般包括故障检测算法,以及必要的数据恢复算法,在正常的程序运行流程中加入故障检测算法,如果检测到程序数据流发生故障,则执行数据恢复流程,如未检测到故障则继续执行正常的程序流程。

常见的软件数据流容错处理机制一般有三变量冗余投票算法,该算法的基本原理为对每个需要加固的数据变量均形成两个冗余变量,并将变量数值赋予另外两个冗余变量,涉及该变量的运算赋值语句均执行三遍冗余,在该数据变量需要进行运算或重新赋值前,将该数据变量与该变量的两个冗余变量进行对比,如数值一致则表明未发生故障,如数值不一致则表明故障已发生,这时采用投票原则将数值一致占多数的作为真值将三个变量重新赋值后进行后续程序流程。三变量冗余投票算法的实质是利用一种软件冗余的方法实现了三硬件冗余的效果,该算法伪代码示例见图2。

Original Code	Modified Code
...	...
x1=x2+x3;	x2'=x2;
...	x2''=x2;
y1=x1;	x3'=x3;
	x3''=x3;
	x1=x2+x3;
	x1'=x2'+x3';
	x1''=x2''+x3'';
	...
	check(x1,x1',x1'');
	y1=x1;

图2 三冗余算法伪代码示例

Fig.2 The pseudo-code example of triple redundancy algorithm

该算法虽然原理简单,执行效果也较好,但是算法实现会产生大量的冗余变量,同时占用大量的寄存器资源并造成较大的性能开销,对资源约束较为严格的空空导弹飞控软件来说并不适用。

本文所采用的基于冗余备份及校验和的软件数据流加固方法是基于冗余复算^[6]思想,利用变量的冗余备份及校验和对

程序数据流故障进行检验及恢复。该方法主要分为故障检测及数据恢复两部分机制,算法具体步骤如下:

(1)首先对某一程序功能模块中所有需要加固的程序变量 v_1 、 $v_2 \cdots v_n$ 初次赋值时进行备份,生成变量副本 v_{1_backup} 、 $v_{2_backup} \cdots v_{n_backup}$ 。

(2)对该模块中所有需要加固的程序变量 v_1 、 $v_2 \cdots v_n$ 及进行累加,形成变量累加值 sum ,对所有变量副本 v_{1_backup} 、 $v_{2_backup} \cdots v_{n_backup}$ 进行累加,形成变量副本累加值 sum_backup 。

(3)在变量 v_1 、 $v_2 \cdots v_n$ 再次引用或赋值前,对变量进行故障检测,即与其副本变量 v_{1_backup} 、 $v_{2_backup} \cdots v_{n_backup}$ 进行对比。

(4)以变量 v_1 进行举例,如果检测到变量 v_1 与其副本变量 v_{1_backup} 数值不一致,则判断该变量发生故障,执行下列数据恢复步骤。

(a)对比变量累加值 sum 及变量副本累加值 sum_backup ,如数值一致则表明 sum 变量数值未发生变化,表明该数值可信,则在此时将所有加固的程序变量 v_1 、 $v_2 \cdots v_n$ 累加后与原累加值 sum 进行对比,如相等则表明变量 v_1 未发生改变, v_1 的数值正确,将该数值重新赋予 v_{1_backup} ,如程序变量 v_1 、 $v_2 \cdots v_n$ 累加后的值与原累加值 sum 不一致,则表明变量 v_1 已发生改变, v_1 的数值错误,此时认为变量副本 v_{1_backup} 中的数值正确,将变量副本 v_{1_backup} 中的正确数值重新赋予变量 v_1 ,同时对变量 sum 及 sum_backup 的数值用变量 v_1 的正确数值进行更新。

(b)如果变量累加值 sum 及变量副本累加值 sum_backup 经对比不一致,则表明 sum 或 sum_backup 发生了变化,则将所有程序变量 v_1 、 $v_2 \cdots v_n$ 累加后与原累加值 sum ,将变量副本 v_{1_backup} 、 $v_{2_backup} \cdots v_{n_backup}$ 累加后与副本变量原累加值 sum_backup 均进行对比,取相等的作为 sum 的真值,重复进行步骤(a)的对比。

(5)如检测到变量与其变量副本数值一致,则判断该变量无故障发生,继续执行后续程序步骤。

(6)程序变量 v_1 、 $v_2 \cdots v_n$ 重新赋值后,将对应的副本变量 v_{1_backup} 、 $v_{2_backup} \cdots v_{n_backup}$ 值进行更新,同时更新累加变量 sum 及其副本累加变量 sum_backup 的数值。

算法原理框图如图3所示。

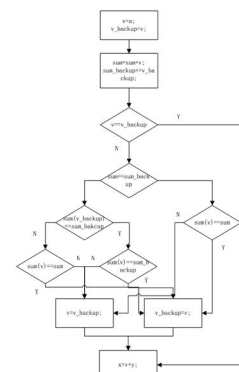


图3 算法原理框图

Fig.3 The algorithm principle block diagram

该算法与常规的三变量冗余投票算法相比,仅需要对变量进行双冗余,而数据错误判断及恢复的算法复杂度并没有增加,因此节省了大量的计算机资源。而且如果三变量冗余投票算法中三变量有两个变量同时发生数值跳变,则该算法会将故障值作为真值进行后续程序运行,不能正确进行故障检测与数值恢复,算法将会失效。而本文所采用的基于冗余备份及校验和的软件数据流加固方法则通过加入副本累加值sum_backup变量,在程序变量v和累加变量sum同时发生数值跳变的时候仍然可以进行正确的故障检测与数值恢复,因此该算法具有较强的鲁棒性。

从空空导弹飞控软件实际运行情况看,不同数据变量对于程序运行的可靠性影响并不相同,只有小部分重要的数据变量对于系统运行的可靠性有重大影响。因此,并不需要对程序中所有的数据变量均进行加固,可综合考虑空空导弹飞控软件的实际特点与计算资源的合理运用,重点识别出对导弹安全性、飞行任务达成影响较大的数据变量,如导弹的位置、速度、姿态、目标信息、传感器信息、过载指令、舵控指令、引信起爆指令等变量,利用本文所述的基于冗余备份及校验和的软件数据流加固算法进行数据加固。

3 试验研究(Research and experiment)

算法的有效性需要试验来进行证明,为了验证本文所提出的基于冗余备份及校验和的软件数据流加固算法的有效性,设计如下试验:利用某型空空导弹数字仿真环境^[7],对飞控软件中识别出的较为重要的数据变量:稳定控制模块中滚转、俯仰和偏航三通道过载指令,利用基于冗余备份及校验和的软件数据流加固算法进行加固。同时在程序中模仿单粒子反转引起的瞬态故障模式,利用软件插装技术在程序不同运行周期对三通道过载指令依次进行故障注入,改变三通道过载指令变量的数值。加固算法和模拟故障代码实现后,运行该数字仿真平台进行试验。本次试验中加固算法及故障注入部分的伪代码示例如图4所示。

Original Code	Modified Code
<pre>if (Gx > 0.5) { Gx_backup = Gx; sum_backup = sum; }</pre>	<pre>if (Gx > 0.5) { Gx_backup = Gx; sum_backup = sum; }</pre>
<pre>if (Gy > 0.5) { Gy_backup = Gy; sum_backup = sum; }</pre>	<pre>if (Gy > 0.5) { Gy_backup = Gy; sum_backup = sum; }</pre>
<pre>if (Gz > 0.5) { Gz_backup = Gz; sum_backup = sum; }</pre>	<pre>if (Gz > 0.5) { Gz_backup = Gz; sum_backup = sum; }</pre>
	<pre>if (Gx_backup != Gx Gy_backup != Gy Gz_backup != Gz) { if (Gx_backup == Gx Gy_backup == Gy Gz_backup == Gz) { sum_backup = sum; } else { sum_backup = Gx_backup + Gy_backup + Gz_backup; } }</pre>
	<pre>if (sum_backup != sum) { sum = sum_backup; }</pre>

图4 伪代码示例

Fig.4 The pseudo-code example

数字仿真平台运行后结果如图5所示。从图中可看出,在程序三个不同运行周期故障注入时,三通道过载指令的故障均被成功识别,算法正确进行了故障检测和数值恢复。



图5 程序运行结果

Fig.5 The result of program

4 结论(Conclusion)

从试验结果看,本文采用的基于冗余备份和校验和的软件数据流加固方法对单粒子翻转引起的数据流软瞬态故障防护效果良好。采用软件防护的方法不依赖于硬件环境,不需要增添或改造硬件,通用性好、成本低且容易实现,应用前景广阔。

本文所采用的试验是在数字仿真环境下进行的,只是初步验证了算法的有效性,而如果要算法真正应用于空空导弹飞控软件,则需要在实际的导弹硬件平台上进行进一步试验。因此,下一步的计划是在某型空空导弹硬件平台上进行试验,综合考虑导弹硬件平台的计算资源及需要加固的程序变量数量,在保证硬件平台机时冗余度的约束条件下,筛选出需要加固的重要数据变量,利用本文采用的基于冗余备份和校验和的软件数据流加固方法进行加固后开展试验,进一步验证算法的有效性,为将算法真正应用于空空导弹中做好准备。

参考文献(References)

- [1] 樊会涛,张鹏鹏.空空导弹面临的挑战[J].航空兵器,2017(2):3-7.
- [2] 傅忠传,陈红松,崔刚,等.处理器容错技术与展望[J].计算机研究与发展,2007,44(1):154-160.
- [3] Oh N,Shirvani P P,McCluskey E J.Error Detection by Duplicated Instructions in Super-Scalar Processors[J].IEEE Trans on Reliability,2002,51(1):63-75.
- [4] Wang C,Kim H,Wu Y,et al.Compiler-Managed Software-Based Redundant Multi-Threading for Transient Fault Detection[C].Proc of the Int'l Symp Code Generation and Optimization,2007:244-258.
- [5] Shye A,Blomstedt J,Moseley T,et al.PLR:A Software Approach to Transient Fault Tolerance for Multi-core Architectures[J].IEEE Trans on Dependable and Secure Computing,2009,6(2):135-148.
- [6] S.K.Reinhardt,S.S.Mukherjee.Transient Fault Detection via Simultaneous Multithreading[C].Proc.of the 27th International Symposium on Computer Architecture,2000:25-36.
- [7] 王明惠,朱光宇,王慧敏.基于PowerPC的飞控软件设计实现及验证[J].航空兵器,2013(4):54-57.

作者简介:

李 伟(1983-),男,硕士,工程师.研究领域:飞行控制软件.