软件可靠性与安全性的区别分析及其证明*)

樊林波[1,2] 吴映程[1] 赵 明[1,3] 代碧锋[4]

(贵州大学贵州省可靠性工程研究中心 贵阳 550001)[1] (遵义师范学院计算机科学系 遵义 563002)[2]

(瑞典耶夫勒大学技术系)[3] (西南大学新闻文化中心 重庆 400715)[4]

摘 要 可靠性和安全性是软件质量中的重要属性[1,2]。虽然文献[3-7],都从不同的角度给出了二者之间的区别和联系,但对这两者之间关系的认识是不一致的。在实践中,如何区分二者的关系还停留在一般性的描述上,没有统一的认识。本文从它们的基本概念着手,重新对二者的基本概念进行抽象描述,进行本质探析后给出二者之间的区别和联系,并对得出的结论给予了证明。

关键词 软件可靠性,软件安全性,形式化描述,可靠性与安全性的关系

Analysis and Proof of the Differences between Software Reliability and Safety

FAN Lin-bo[1,2] WU Ying-cheng[1] ZHAO Ming[1,3] DAI Bi-feng[4]

(Reliability Engineering Center of Guizhou Province, Guizhou University, Guiyang 550001, China)[1]

(Dept of Computer Science, Zunyi Normal College, Zunyi 563002, China)[2]

(Department of Technology, University of Gävle, Sweden)[3] (The News culture center of Xinan University, Chongqing 400715, China)[4]

Abstract Reliability and safety are the most important measures in software quality[1,2]. Although[3-7] propose the differences and relationship between them in different views, there has clearly been the inconsistence about the cognitions on the their relationship. In practice, there have been no consensus except for general descriptions. This paper provides an abstract description on the concepts, and then explores the differences between the reliability and safety. Based on the formal definition on the concepts, the relationship between the measures of software reliability and safety is theoretically proved in this paper.

Keywords Software reliability, Software safety, Formal method, Relationship between reliability and safety

1 引言

随着软件在尖端领域和大型复杂系统中的广泛应用,软件的可靠性和安全性是"用户"非常关心的问题,它的好坏不仅关系到系统的成败,还涉及到国家和人民生命财产的安全。但是,软件的可靠性和安全性的研究一直是软件系统中的一个难题,几十年来,很多国家都将该领域的研究作为科技攻关重点,如英国的 AIVEYC(软件可靠性和度量标准)计划,欧洲的 ESPRIT(欧洲信息技术研究与发展战略)计划[8],SPMMS(软件生产和维护保障)课题[9],EUREKA(尤里卡)计划[10]等。遗憾的是至今都没有取得突破性的进展。什么样的软件是可靠的,什么样的软件是安全的,至今没有很好的定量描述方法,甚至连二者之间究竟是什么样的关系,它们之间的区别在何处,联系在何处,不同的文献都有不同的看法,如文献[6]指出,安全性是可靠性的子集,认为不安全的系统一定不是可靠的;文献[3]认为可靠性是安全性的子集,认为可靠的系统并一定是安全的,可靠性只是安全性的一个方面。在实践中也没有一个统一的区分标准,在很多情况下,人们都将二者混为一谈,作为软件质量中二个重要的质量属性,准确刻画二者之间的区别和联系,既是必要的也是现实的。为了准确理解二者的关系,本文先对软件可靠性和软件安全性的基本概念作简单介绍,然后用形式化方法对二者涉及的有关概念进行了重新描述,最后给出了二者之间准确的区别和联系,并对此结论进行了证明。

2 软件可靠性和安全性概念介绍

可靠性的概念给出比较多,在文献[11-14]里都分别给出了不同的描述,但人们习惯上是这样定义软件可靠性的:软件在规定的时间内,规定的条件下,完成规定功能的概率。该定义给出了三个规定,首先是规定的时间,这里的时间既可理解为软件的生命周期,也可以理解为软件执行中的某一段时间,还可以理解为软件在执行中真正起作用的时间,这要根据研究需要来确定;其次是规定的条件,规定的条件一般是指能明确告诉的与软件运行相关的各种因素,相关的因素既有内部的也有外部的,但都是人们已知的会对执行功能有关的因素;最后讲执行规定的功能,所谓规定的功能就是人们主观上的期望功能,或者说是人们对软件给出的理想功能,实际执行中是否达到这理想目标,有多大的概率能达到这个目标,就是可靠性研究要完成的任务。

安全性的定义目前还没可靠性那样明确,但文献[15-18]都是按这样来理解的:软件的安全性是指软件在规定的运行时间内是否会对系统本身和系统外界造成危害概率,这种危害包括人身安全、重大财产损失和人们极不期望发生的事件等。安全性并不强调系统的功能,关注的是系统哪些地方是

*)国家自然科学基金项目(60473054)。**樊林波** 博士生,讲师,研究方向为软件可靠性和安全性、软件测试;**吴映程** 硕士研究生,研究方向为软件可靠性;**赵 明** 教授,博士生导师,研究方向为软件工程、可靠性工程、软件安全。

脆弱的和哪些地方受到潜在威胁。虽然从概念上安全性不考虑系统的功能，但安全事故的发生也是有条件的，这些条件可能来自系统内部，也可能来自系统外部，有些条件可能是人们未知的，也可能是人们已知的，但没有引起重视。总之，安全事故的产生是系统没有达到人们期望的结果，也没有完成人们给系统规定的功能。

通过对概念的简单分析，也可以得出以下几点结论：一是可靠性和安全性都是在一定的时间范围内来考虑的；二是不可靠和不安全的系统，都是系统运行中产生状态和理想状态（人们希望的状态）之间出现了差距；三是不可靠的系统和不安全的系统都是人们不希望发生的，它们表现的结果都是人们不可接受的，只是不接受的程度有区别。下面通过对二者概念的形式描述，来详细刻画两者的区别和联系。

3 软件可靠性和安全性的形式描述

定义 1（理想状态系统） 系统在规定的时间内，各时刻所表现的状态都是人们所希望的状态，由这样的状态所组成的系统，用符号 S 表示。

定义 2（受限状态系统） 系统在规定的时间内，在规定的条件下，各时刻所表现出来的状态所组成的系统，用符号 $S[R]$ 表示。

定义 3（真实状态系统） 系统在规定的时间内，在真实条件下，各时刻所表现出来的状态所组成的系统，用符号 $S[A]$ 表示。注：这里的真实条件和定义 2 的规定条件是有区别的，真实条件包括人们已知的和未知的条件，规定的条件往往是人们已知的条件。

断言 1：假设 r 是 $S[R]$ 中的条件个数，a 是 $S[A]$ 中的条件个数，则 $r \leqslant a$。

软件实际执行过程中，由于所面临的条件不同，其表现出来的状态就可能不同，这些条件可能是人们已知的或已经探测出来的，但有些也可能是人们没有注意到的或是未知的，在 $S[R]$ 中，人们是在规定的条件（或者已知的条件）下来思考系统将出现的状态，即是在 r 个条件下预计系统将出现的状态，这 r 是已知的。但在 $S[A]$ 中，系统的状态是在真实条件中所表现出来的，这真实条件既有已知的 r 个，还可能存在未知的无数个，故有 $r \leqslant a$。

定义 4（环境） 环境就是可能导致系统状态改变的各种条件，我们用符号 E 来表示。E 可以是日常生活中的如温度、湿度、压力等，也可以是与 S 发生作用的其他系统；不同的环境用下标区别，如 $E1$，$E2$ 分别表示由条件 1 和条件 2 构成的不同的环境；用 $E[R]$ 表示有 r 个条件构成环境。$E[R]=\{E1,E2,\cdots,Er\}$。

断言 2：假设 $E[R]$ 是 $S[R]$ 中的环境，$E[A]$ 是 $S[A]$ 中的环境，则 $E[R] \subseteq E[A]$。

由定义可知 $S[R]$ 中 $E[R]$ 的个数是 r 个，即 $E[R]=\{E1,E2,\cdots,Er\}$；$S[A]$ 中 $E[A]$ 的个数是 a 个，则 $E[A]=\{E1,E2,\cdots,Ea\}$。由断言 1 知，$r \leqslant a$，所以 $E[R] \subseteq E[A]$。

定义 5（系统状态） 系统在执行过程中在某个时间和某种环境下表现出来的各种参量的组合，用符号 P 表示，不同时刻的状态我们用下标区别，如 P_{t0}，P_{t1} 就分别表示系统在 $t0$ 时刻和 $t1$ 时刻两个不同的理想状态；$P_{t0}[1]$，$P_{t1}[1]$ 分别表示系统在 $t0$ 时刻和 $t1$ 时刻，在环境 $E1$ 下表现出来的两个不同的状态。

定义 6（可能状态 $P\bigstar E$） $P\bigstar E$ 表示系统在某种环境 E 下可能表现的系统状态。规定 $P\bigstar \Phi=P$；$P\bigstar Er=(P,P[r])$；$P\bigstar E[R]=(P,P[1],P[2],\cdots,P[r],P[1,2],\cdots,P[r-1,r],P[1,2,3],\cdots,P[i,j,k],\cdots,P[i,j,\cdots,r]$，其中在同一个 [ ] 的数满足 $1 \leqslant i<j<k<\cdots<r$）。

定义 7（偏离代价） 系统 S 在 E 下所表现出来的状态 $P[E]$ 与系统 S 在该时刻的理想状态 P 之间产生的偏离，并由此偏离带来的损失，称为偏离代价。用符号 C 表示，C 既可理解为日常生活中的金钱，也可以理解为市场份额、声誉影响和满意度等。不同时间和不同状态的偏离损失，用 C 的其他标记来区别，如 $C_{t0}[1,2]$ 表示在 $t0$ 时刻状态 $P_{t0}[1,2]$ 与 P_{t0} 的偏离代价。

定义 8（可接受的最大代价） 表示在某时刻 t，人们可接受的系统的最大偏离代价值就称可接受的最大代价，用符号 C_t 表示，不同时刻的最大代价时间下标区别，如 C_{t0} 表示在 $t0$ 时刻可接受的最大代价。该值主要用来判定系统在某时刻的状态是否可靠。

定义 9（安全事故的最小代价） 在某时刻 t，人们用来判定系统是否产生安全事故的最小代价值，就称为安全事故的最小代价，用符号 C_A 表示，不同时刻用下标区别，如 C_{A0} 表示在 $t0$ 时刻界定安全事故的最小代价。

断言 3：一般地，系统在某时刻 t 规定的 C_t 和 C_A 之间满足 $C_t \leqslant C_A$。

系统的同一时间里，如果 $C_t \leqslant C_A$ 不成立，则必有 $C_t > C_A$ 成立，这就说明安全事故造成的代价是在人们可接受的范围内，人们既可接受，就证明危害不大，那就谈不上安全事故了，这与安全事故本身的定义矛盾。所以 $C_t \leqslant C_A$。

定义 10（系统可靠性） 假定系统在规定的 T 时间内，在规定的 $E[R]$ 中，所有可能呈现的状态 $P\bigstar E[R]$ 与对应的理想状态 P 之间的偏离代价为 $C[R]$，则称系统在 T 时间内是可靠的当且仅当 $\forall(C_t[R]) \leqslant C_t)=1$。

定义 11（系统安全性） 假定系统在规定的 T 时间内，在真实环境 $E[A]$ 中，所有可能呈现的状态 $P\bigstar E[A]$ 与对应的理想状态 P 之间的偏离代价 $C[A]$，则称系统在 T 时间内是安全的当且仅当 $\forall(C_t[A] \leqslant C_A)=1$。

定义 12（系统可靠性分析） 设 $E[R]$ 为系统规定的环境，其对应的系统为 $S[R]$，则对 $S[R]$ 的可靠性分析包含三个任务：一是找出 $S[R]$ 所有可能出现的状态 $P\bigstar E[R]$；二是确定每个 $P\bigstar E[R]$ 与对应的理想状态之间产生的偏离损失 $C[R]$；三是求 $\forall(C_t[R] \leqslant C_t)$ 的真假值。经这三步便可以确定系统是否是可靠的。

定义 13（系统安全性分析） 设 $E[A]$ 为系统的真实环境，其对应的系统为 $S[A]$，则对 $S[A]$ 的安全性分析包含三个任务：一是找出 $S[A]$ 所有可能出现的状态 $P\bigstar E[A]$；二是确定每个 $P\bigstar E[A]$ 与对应的理想状态之间产生的偏离损失 $C[A]$；三是求 $\forall(C_t[A] \leqslant C_t)$ 的真假值。经这三步便可以确定系统是否是安全的。

4 软件可靠性和安全性的关系及证明

定理 1 如果系统在规定的时间 T 内，所面临的真实条件 $E[A]$ 和系统规定的条件 $E[R]$ 完全相同，即有 $E[R]=E[A]$，则系统是可靠的就可判定系统是安全的，反之不成立（即系统是安全的不能得出系统是可靠的）。

证明：假设有一系统 S，在时间 T 内的理想状态序列为 $P[T]=\{P_{t0},P_{t1},P_{t2}\}$，如果将 S 限定在规定的环境 $E[R]=$

$\{E1,E2\}$中，得到新的系统 $S[R]$，下面我们对 $S[R]$作可靠性分析。

① 求 $P\bigstar E[R]$，由定义 5 可知，$P\bigstar E[R]=\{P_{t0}\bigstar E[R],P_{t1}\bigstar E[R],P_{t2}\bigstar E[R]\}=\{(P_{t0}[1],P_{t0}[2],P_{t0}[1,2]),(P_{t1}[1],P_{t1}[2],P_{t1}[1,2]),(P_{t2}[1],P_{t2}[2],P_{t2}[1,2])\}$。即 $S[R]$有九种可能的状态。

② 求对应状态的偏离代价，由定义 7，我们可知 $C[R]=\{(C_{t0}[1],C_{t0}[2],C_{t0}[1,2]),(C_{t1}[1],C_{t1}[2],C_{t1}[1,2]),(C_{t2}[1],C_{t2}[2],C_{t2}[1,2])\}$。

③ 由假设 $S[R]$是可靠的，即有 $\forall(C_t[R]\leqslant C_t)=1$，可知$(C_{t0}[1]\leqslant C_{t0})\wedge(C_{t0}[2]\leqslant C_{t0})\wedge(C_{t0}[1,2]\leqslant C_{t0})\wedge(C_{t1}[1]\leqslant C_{t1})\wedge(C_{t1}[2]\leqslant C_{t1})\wedge(C_{t1}[1,2]\leqslant C_{t1})\wedge(C_{t2}[1]\leqslant C_{t2})\wedge(C_{t2}[2]\leqslant C_{t2})\wedge(C_{t2}[1,2]\leqslant C_{t2})=1$。

下面我们对 S 所处的真实系统 $S[A]$作安全性分析。

① 求 $P\bigstar E[A]$，有假设 $E[R]=E[A]$，由定义 5 可知，$P\bigstar E[A]=\{P_{t0}\bigstar E[A],P_{t1}\bigstar E[A],P_{t2}\bigstar E[A]\}=\{(P_{t0}[1],P_{t0}[2],P_{t0}[1,2]),(P_{t1}[1],P_{t1}[2],P_{t1}[1,2]),(P_{t2}[1],P_{t2}[2],P_{t2}[1,2])\}$。即 $S[A]$也有九种可能的状态，且和 $S[R]$是完全相同的。

② 求对应状态的偏离代价，由定义 7，我们可知 $C[A]=\{(C_{t0}[1],C_{t0}[2],C_{t0}[1,2]),(C_{t1}[1],C_{t1}[2],C_{t1}[1,2]),(C_{t2}[1],C_{t2}[2],C_{t2}[1,2])\}$。

③ 判断 $S[A]$是否是安全的，由断言 3(有 $C_t\leqslant C_A$)并结合以上可靠性分析的第③步的结论，可得出$(C_{t0}[1]\leqslant C_{A0})\wedge(C_{t0}[2]\leqslant C_{A0})\wedge(C_{t0}[1,2]\leqslant C_{A0})\wedge(C_{t1}[1]\leqslant C_{A1})\wedge(C_{t1}[2]\leqslant C_{A1})\wedge(C_{t1}[1,2]\leqslant C_{A1})\wedge(C_{t2}[1]\leqslant C_{A2})\wedge(C_{t2}[2]\leqslant C_{A2})\wedge(C_{t2}[1,2]\leqslant C_{A2})=1$。为此有 $\forall(C_t[A]\leqslant C_A)=1$ 成立，即 $S[A]$是安全的。

由以上就证明了当 $E[R]=E[A]$时，系统是可靠的就可判定系统是安全的。

下面我们简单说明当 $E[R]=E[A]$时，为什么有系统是安全的不能判定系统是可靠的。

由以上的分析已经知道，$S[A]$和 $S[R]$之间的状态和对应的偏离代价都是相同的，由假设 $S[A]$是安全的，根据定义 11，有 $\forall(C_t[A])\leqslant C_A)=1$，即$(C_{t0}[1]\leqslant C_{A0})\wedge(C_{t0}[2]\leqslant C_{A0})\wedge(C_{t0}[1,2]\leqslant C_{A0})\wedge(C_{t1}[1]\leqslant C_{A1})\wedge(C_{t1}[2]\leqslant C_{A1})\wedge(C_{t1}[1,2]\leqslant C_{A1})\wedge(C_{t2}[1]\leqslant C_{A2})\wedge(C_{t2}[2]\leqslant C_{A2})\wedge(C_{t2}[1,2]\leqslant C_{A2})=1$，但是根据断言 3($C_t\leqslant C_A$)，由 $C_t[A]\leqslant C_A$ 为真，不能得出 $C_t[A]\leqslant C_t$ 为真的结论。因 $C_t[E]=C_t[A]$，所以就不能得出 $\forall(C_t[R]\leqslant C_t)=1$ 的结论。故当 $E[R]=E[A]$时，系统是安全的不能判定系统是可靠的。

推论 1 在定理 1 的条件下，当 $C_t=C_A$ 时，系统是安全的就可判定系统是可靠的。

证明：由以上定理 1 的证明过程可知，$S[A]$和 $S[R]$之间的状态 $P\bigstar E[A]=P\bigstar E[R]$，和对应的偏离代价 $C[A]=C[R]$，如果系统是安全的，则有 $\forall(C_t[A]\leqslant C_A)=1$，由 $C[A]=C[R]$和 $C_t=C_A$，即可得出 $\forall(C_t[R]\leqslant C_t)=1$，即可判定系统是可靠的。

定理 2 设 $E[R]$是系统规定的环境，$E[A]$是系统的真实环境，如果 $E[R]\subset E[A]$，则 $P\bigstar E[R]\subset P\bigstar E[A]$，$C[R]\subset C[A]$。

证明：设有一系统 S，在时间 T 内的理想状态为 $P[T]=\{P_{t0},P_{t1}\}$，假定 $E[R]=\{E1,E2\}$，$E[A]=\{E1,E2,E3\}$，下面我们对 $S[R]$作可靠性分析和对 $S[A]$作安全性分析。

①对 $S[R]$的可靠性分析得出

$P\bigstar E[R]=\{P_{t0}\bigstar E[R],P_{t1}\bigstar E[R]\}=\{(P_{t0}[1],P_{t0}[2],P_{t0}[1,2]),(P_{t1}[1],P_{t1}[2],P_{t1}[1,2])\}$

$C[R]=\{(C_{t0}(1),C_{t0}(2),C_{t0}(1,2)),(C_{t1}(1),C_{t1}(2),C_{t1}(1,2))\}$

②对 $S[A]$的安全性分析得出

$P\bigstar E[A]=\{P_{t0}\bigstar E[A],P_{t1}\bigstar E[A]\}=\{(P_{t0}[1],P_{t0}[2],P_{t0}[1,2],P_{t0}[1,3],P_{t0}[2,3],P_{t0}[1,2,3]),(P_{t1}[1],P_{t1}[2],P_{t1}[1,2],P_{t1}[1,3],P_{t1}[2,3],P_{t1}[1,2,3])\}$

$C[A]=\{(C_{t0}(1),C_{t0}(2),C_{t0}(1,2),C_{t0}(1,3),C_{t0}(2,3),C_{t0}(1,2,3)),(C_{t1}(1),C_{t1}(2),C_{t1}(1,2),C_{t1}(1,3),C_{t1}(2,3),C_{t1}(1,2,3))\}$

通过比较我们看出 $P[R]$和 $C[R]$都分别包含在 $P[A]$和 $C[A]$中，

即 $P[A]=P[R]+\{(P_{t0}[1,3],P_{t0}[2,3],P_{t0}[1,2,3]),(P_{t1}[1,3],P_{t1}[2,3],P_{t1}[1,2,3])\}$

$C[A]=C[R]+\{(C_{t0}(1,3),C_{t0}(2,3),C_{t0}(1,2,3)),(C_{t1}(1,3),C_{t1}(2,3),C_{t1}(1,2,3))\}$

所以有 $P\bigstar E[R]\subset P\bigstar E[A]$，$C[R]\subset C[A]$。

推论 2 在定理 2 的条件下，如果 $C_t=C_A$，则系统是安全的就可判定系统是可靠的，反之不成立(即系统是可靠的不能判定系统是安全的)。

证明：设 $E[R]$是系统规定的环境，$E[A]$是系统的真实环境，$E[R]\subset E[A]$，如果 $S[A]$是安全的，则有 $\forall(C_t[A]\leqslant C_A)=1$，由已知条件 $C_t=C_A$，可得 $\forall(C_t[A]\leqslant C_t)=1$。根据定理 2 的结论 $C[R]\subset C[A]$，则有 $\forall(C_t[R]\leqslant C_t)=1$，由此证出系统 $S[R]$是可靠的。

反之不成立，是因为由 $C_t=C_A$，和可靠性条件 $\forall(C_t[R]\leqslant C_t)=1$，我们可得 $\forall(C_t[R]\leqslant C_A)=1$，但是定理 2 的结论 $C[R]\subset C[A]$，不能得出 $\forall(C_t[A]\leqslant C_A)$为真的结论，故不能判定系统 $S[A]$是安全的。

结束语 本文通过对系统相关因素的形式描述，很清晰地刻画了可靠性和安全性之间的区别和联系。文章可从以下几个方面更容易理解。一是不可靠和不安全对外界造成的损失或代价是不一样的，这是人们从影响结果上对二者界定；二是造成这些损失的原因是系统的实际状态与人们希望的状态不一样或者说偏离；三是这种偏离是由与系统相关的环境(即各种因素)造成的；四是人们在分析可靠性和安全性从不同的立场或角度来考虑与它们有关的环境，这环境有时候有区别，有时候又相同，这就导致二者关系的不确定性；五是安全性和可靠性的关系就是环境、偏离代价和界定值三者之间的动态关系。

另外，现实的环境往往比较复杂，有些是我们未知的，有些是我们已知的，有些还是我们已知但没有引起重视的，这就是安全性分析往往比可靠性分析更困难的地方，这也是凡提到安全性的研究往往都要涉及到安全风险分析[19,20]，那是因为安全风险分析可更全面地考虑环境因素(与系统相关的因素)，对可靠性的研究是规定环境后，只考虑系统的功能。

参 考 文 献

[1] Yamada S, Tokuno K, Kasano Y. Quantitative Assessment Models for Software Safety/Reliability. Electronics and Communications in Japan, Part2, 1998, 81(5)

[2] Mohagheghi P, Qualit R C. Prduetivity and Economic Benefits of

Software Reuse:a Review of Industrial Studies. Empir Software Eng,2007,12:471-516
[3] Leveson N G. Software Safety: Why, What, and How. ACM Computing Surveys,1986,18(2):125-163
[4] Tokuno K, Yamada S. Stochastic Software Safety/reliability Measurement and Its Application. Annals of Software Engineering,1999(8):123-145
[5] Yamada S. Software Reliability/Safety Assessment. J. Japan Society for Safety Engineering,1990,33(6):432-441
[6] Musa J D. Software reliability engineering. McGraw-Hill,1999
[7] Abbott R J. Resourceful Systems for Fault Tolerance, Reliability, and Safety. ACM Computing Surveys,1990,22(1)
[8] Fan I, Filos E. Concurrent Engineering Projects Supported by The European Commission's ESPRIT Programme and Future Trends. Concurrent Engineering-Research and Applications, 2001,9(2):166-173
[9] Hurst R. SPMMS-Information Structures in Software Management. Software Engineering Journal,1986,1(1):50-57
[10] Barker K, Dale A, Geroghio L. Management of Collaboration in EUREKA Projects: Experiences of UK Participant Technology Analsys & Strategic Management,1996,8(4):467-482
[11] Lyu M R. Handbook of software reliability engineering. McGraw-Hill,1996
[12] Bodsberg L, Hokstad P. Transparent Reliability Model for Fault-Tolerant Safety System. Reliability Engineering and System Safety,1997:25-38
[13] Ramamoorty C V, Bastani F B. Software Reliability-Status and Perspective. IEEE Ttans. on Software Engineering, 1982, SE-8 (4)
[14] Yamada S. Software reliability model—Fundamentals and applications. Nikka Giren,1994
[15] Gwandu B A L, Creasey D J. Using Formal Methods in Design for Reliability as Applied to An Electronic System That Integrates Software and Hardware to Perform a Function. Microelectron Reliab. ,1995,35(8):1111-1124
[16] Leveson N G. Software System Safety and Computers. Addison-Wesley Publishing Company, Inc,1995
[17] Keene SJ Jr. Assuring software safety//Proc. IEEE Annual Reliability Maintenance Symp. Las Vegas,1992:274-279
[18] Yamada S. Reliability/safety evaluation of software. Safety Eng, 1994,33:432-441
[19] Kang H G, Sung T. An Analysis of Safety-Critical Digital Systems For Risk-Informed Design. Reliability Engineering and System Safety,2002,78:307-314
[20] Cai K Y. System Failure Engineering and Fuzzy Methodology in Introductory Overview. Fuzzy Sets and Systems, 1996, 83: 113-133

(上接第 251 页)

方法对于低频正弦噪音的去除效果最好,而采用了低频抑制的加权切比雪夫方法最差。对其它低频噪音的实验结果与此类似,因此不再赘述。

表 3 重建误差比较:添加正弦低频噪音

K	5	10	15	20	25	30
W-Cheby	0.3	0.21	0.16	0.17	0.22	0.23
W-Cheby2	0.34	0.3	0.26	0.27	0.29	0.29
Bi-Cheby	0.2	0.19	0.1	0.11	0.11	0.11

注:W-Cheby2 表示采用低频抑制的加权切比雪夫方法。

综上所述,对于含有噪音的移动对象轨迹,双切比雪夫方法能够在相同压缩比下更好地抵抗噪音,用双切比雪夫系数重建的轨迹与真实轨迹更加接近。这是因为,双切比雪夫近似模型综合考虑了位置曲线的误差和速度曲线的误差。通过双切比雪夫距离的约束,位置序列和速度序列之间的导数关系对二者的噪音产生中和作用,从而在一定程度上达到抗噪目的。

结束语 本文针对移动对象轨迹的压缩、近似问题提出了双切比雪夫方法。该方法的主要贡献是:

• 突破了现有方法只能处理单一信息的限制,综合位置和速度信息,建立了二次优化模型。

• 利用两类切比雪夫多项式与移动对象轨迹的导数对应关系,推出了模型的理论解。

• 提出了双切比雪夫系数的快速数值算法,使得计算一个系数的复杂度降为线性。

• 通过实验比较证实了双切比雪夫方法能在压缩轨迹数据的同时更有效地抑制噪音,使重建的轨迹在欧氏距离意义下更加接近真实轨迹。

在进一步的工作中,我们考虑将双切雪夫方法应用到不含速度信息的轨迹采样数据中。一种可能的途径是,用位置序列拟合出一条近似的速度序列,然后再针对这两个序列使用双切比雪夫方法。

参 考 文 献

[1] Ding Zhiming, Güting R H. Managing Moving Objects on Dynamic Transportation Networks// Proc. the 16th Int. Conf. Science and Statistical Database Management. Santorini, Greece, 2004:287-296
[2] Faloutsos C, Ranganathan M, Manolopoulos Y. Fast subsequence matching in time-Series databases// Proc. the Int. Conf. Management of Data. Minneapolis, Minnesota, USA,1994:412-429
[3] Rafiei D, Mendelzon A. Similarity-based queries for time series data// Proc. the Int. Conf. Management of Data. Tucson, Arizona, USA,1997:13-25
[4] Chan K P, Fu A W C. Efficient time series matching by wavelets //Proc. the 15th Int. Conf. Data Engineering. Sydney, Australia, 1999:126-133
[5] Chakrabarti K, Garofalakis M, Rastogi R, et al. Approximate query processing using wavelets// Proc. the 26th Int. Conf. Very Large Data Bases Conference. Cairo, Egypt,2000:111-122
[6] Guha S, Harb B. Approximation algorithms for wavelet transform coding of data streams// Proc. the 17th Annual ACM-SIAM Symposium on Discrete Algorithm. Miami, Florida, USA, 2003:698-707
[7] Cai Y H, Ng R. Indexing spatio-temporal trajectories with Chebyshev polynomials// Proc. Int. Conf. Management of Data. Paris, France,2004:599-610
[8] Keogh E, Chakrabarti K, Pazzani M, et al. Dimensionality reduction technique for fast similarity search in large time series databases. Journal of Knowledge and Information Systems, 2001, 3 (3):263-286
[9] Keogh E, Chakrabarti K, Pazzani M, et al. Locally adaptive dimensionality reduction for indexing large time series databases //Proc. Int. Conf. Management of Data. Santa Barbara, California, USA,2001:151-162
[10] Press W H, Flannery B P, Teukolsky S A, et al. Numerical Recipes: The Art of Scientific Computing. 2nd Edition. Cambridge University Press,1994
[11] Mason J C, Handscomb D. Chebyshev Polynomials. Chapman & Hall,2003