文章编号:1674-8190(XXXX)XX-001-14

# 基于AADL2SPN的飞行控制系统可靠性分析

罗文斌,陆中,程大炜,缪炜润
(南京航空航天大学 民航学院,南京 211106)

**摘 要:** 飞行控制系统是典型的安全关键系统,其可靠性对保证飞机安全运行具有重要作用。传统可靠性分析方法过于依赖分析人员的经验,主观性强,极易导致可靠性模型与设计模型之间存在不一致性。综合利用架构分析设计语言(AADL)和随机Petri网(SPN)描述系统的故障传播行为,提出一种基于模型的可靠性分析方法;利用AADL构建某横侧向电传飞控系统的名义模型和错误模型,提出AADL模型中错误传播相关信息的提取方法,利用所提取信息自动生成描述系统故障传播行为的SPN模型;在此基础上,通过蒙特卡洛仿真完成该横侧向电传飞控系统的可靠性评估。经与故障树分析方法对比,结果表明:最大相对误差小于0.018%,满足工程需要;所构建的可靠性模型由AADL模型自动生成,能够确保可靠性模型与设计模型的一致性,避免了对设计人员经验的依赖。

**关键词:** 系统可靠性;基于模型的设计;AADL;随机Petri网;飞行控制系统
**中图分类号:** V240.2 **文献标识码:** A
**DOI:** 10.16615/j.cnki.1674-8190.XXXX.XX.01

# Reliability analysis of flight control system based on AADL2SPN

LUO Wenbin, LU Zhong, CHENG Dawei, MIAO Weirun
(College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China)

**Abstract:** Flight control system is typical safety critical system, and the reliability of flight control system plays an important role in ensuring the safe operation of aircraft. Traditional reliability analysis methods have a heavy reliance on the experience of analysts, which makes it easy for inconsistencies between reliability models and design models. The fault propagation behavior of the system is thoroughly described by the Architecture Analysis Design Language (AADL) and stochastic Petri nets (SPN), and a method for model-based reliability analysis is proposed. The nominal model and error model of a lateral fly-by-wire flight control system were constructed using AADL. A method for extracting error propagation information from the AADL model was proposed, and the SPN model described the fault propagation behavior of the system was automatically generated by extracting the information of AADL model. Based on the SPN model, Monte Carlo simulation was used to evaluate the reliability of the lateral fly-by-wire flight control system, compared with the fault tree analysis method, the error is less than 0.018%, which can be neglected in practice. Through the method of this study, the reliability model is automatically generated by the AADL model, which ensures the consistency between the reliability model and the design model and avoids reliance on the experience of designers.

**Key words:** system reliability; model-based design; AADL; SPN; flight control systems

**收稿日期:** 2023-05-31; **修回日期:** 2023-10-12
**基金项目:** 国家自然科学基金(U1733124); 民航安全能力建设基金(2021-196); 航空科学基金(20180252002)
**通信作者:** 陆中, luzhong@nuaa.edu.cn
**引用格式:** 罗文斌, 陆中, 程大炜, 等. 基于AADL2SPN的飞行控制系统可靠性分析[J]. 航空工程进展, XXXX, XX(XX): 1-14.
LUO Wenbin, LU Zhong, CHENG Dawei, et al. Reliability analysis of flight control system based on AADL2SPN[J]. Advances in Aeronautical Science and Engineering, XXXX, XX(XX): 1-14. (in Chinese)

## 0 引 言

飞行控制系统是典型的安全关键系统，其故障或失效将对飞机的运行安全产生严重影响。目前，机载系统和设备的可靠性评估主要依据SAE ARP4761标准[1-2]，其主要使用依赖图分析(DDA)、故障树分析(FTA)、马尔可夫分析(MA)、失效模式和影响分析(FMEA)等传统方法，这些可靠性方法过于依赖分析人员的经验，当可靠性分析人员对设计方案的理解存在偏差时，极易导致设计模型与可靠性模型不一致，使得系统的故障逻辑关系无法被准确描述。现代飞机的飞行控制系统是由机械、电子、电气、液压等单元组成的高集成复杂系统，具有自动化与集成化程度高、组成部分之间交联关系复杂等特点，导致传统可靠性方法的缺点愈发明显。

为弥补传统可靠性分析方法的不足，众多研究人员提出了基于模型的可靠性分析方法，通过形式化模型描述系统架构及其错误行为，以提升分析的一致性和准确性，并降低研制费用[3]。在基于模型的可靠性分析中，常用建模工具包括Simulink[4-5]、AltaRica[6-7]、SysML语言[8]和AADL语言[9-11]等，利用这些工具可以针对系统架构及系统间故障传播关系建立形式化模型，在此基础上自动生成故障树、事件树等安全性模型，从而避免了模型的不一致性。Simulink工具主要用于描述微分方程、传递函数、状态方程等动力学模型，在Simulink模型上通过故障注入分析系统的输出响应，可确定系统故障之间的逻辑关系[12-13]。AltaRica、SysML语言和AADL语言等工具主要用于体系结构模型的构建，利用建模语言描述系统故障逻辑，从而实现可靠性的自动分析[14-15]。

AADL通过定义多种构件来描述系统架构[16]，利用拓展的错误附件来描述系统和部件的错误行为，是针对机载嵌入式系统的专用建模语言，具有强大的建模能力。但是AADL语言本身并不支持安全性分析，需要将AADL模型转换为故障树、马尔可夫模型[17]、时间自动机模型[18]和Petri网[19-20]等安全性分析模型。当前研究主要集中在AADL模型到安全性分析模型的转换规则上，而具体安全性分析则主要依赖现成的安全性分析工具。

本文针对某横侧向电传飞控系统，提出一种基于AADL模型的可靠性分析方法，给出从AADL模型到SPN模型的转换规则，并基于得到的SPN模型，利用蒙特卡洛仿真实现电传飞控系统可靠性评估。

## 1 基于AADL的电传飞控系统建模

基于模型的可靠性分析方法首先需要建立分析模型，本节将利用AADL建立飞控系统模型，包括表示无故障行为的名义模型和表示错误行为的错误模型。

### 1.1 系统名义模型构建

AADL中构件分为三类，分别是软件构件、硬件构件和系统构件，常用各类构件分类和含义如表1所示。软件构件是执行系统行为的构件，是系统中产生数据或处理传输数据的构件；硬件构件是为软件构件提供运行保障的实体；系统构件用于封装软件构件和硬件构件，从而描述系统的架构。

表1 AADL构件及其含义

Table 1 AADL components and the meanings

| 类别 | 构件 | 含义 |
|---|---|---|
| 软件构件 | Data | 表述整型、布尔型等不同类型的数据 |
| | Thread | 并行执行的可调度单元 |
| | Process | 调度、执行线程的虚拟处理器 |
| 硬件构件 | Processor | 调度、执行线程的虚拟处理器 |
| | Memory | 用于存储代码和数据的构件 |
| | Device | 表示与外部环境接口的传感器、作动器或者其他部件 |
| 系统构件 | System | 软件、硬件和其他系统构件集成的单元 |
| | Abstract | 虚拟构件，代表任何构件类型 |

本文所研究的飞控系统由飞行控制计算机系统、舵面驱动机构、传感器系统和操纵面组成。其主要功能是在飞行过程中，采集飞行员发出的指令和传感器的输入信号，经过飞控计算机处理后，通过舵面驱动机构作动操作面，系统架构如图1所示。

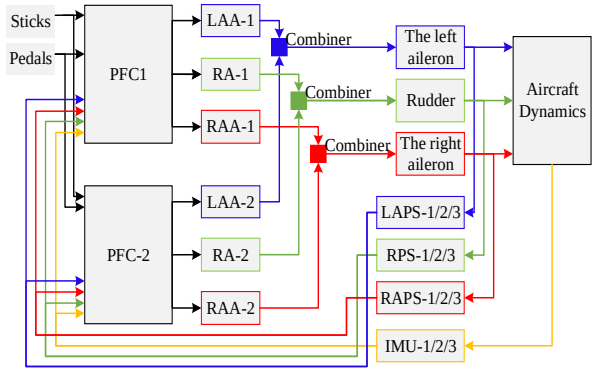



图 1　横侧向电传飞控系统架构

Fig. 1　The architecture of the lateral-directional flight control system

飞行控制计算机子系统具有双冗余架构,由两个相同的主飞行计算机(PFC)组成,每个计算机都有两个冗余的通道。两台计算机架构基本相同,其接收传感器数据,进行指令计算与输出,从而控制飞机的飞行姿态。在一台计算机故障后,通过内部余度管理算法,PFC 仍能由余度间的表决管理输出正确数据,屏蔽故障的负面效应,保障飞行安全。舵面驱动机构包括左副翼作动器(LAA)、右副翼作动器(RAA)和方向舵作动器(RA)。传感器系统包括右副翼位置传感器(RAPS)、左副翼位置传感器(LAPS)、方向舵位置传感器(RPS),它们都具有三重模块化冗余架构,同样三重模块化冗余架构的惯性测量单元(IMU)计算飞机的角度变化。操纵面包括左副翼(Left_aileron)、右副翼(Right_aileron)和方向舵(Rudder)。飞控系统各部件与 AADL 构件的对应关系如表 2 所示。

表 2　飞控系统部件与 AADL 构件对应关系

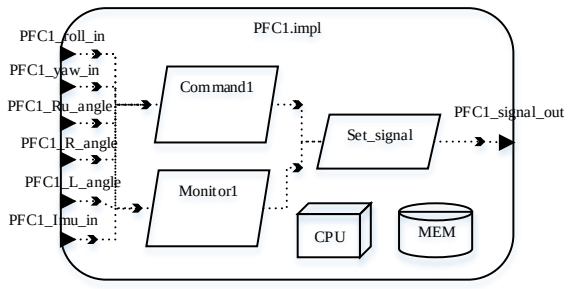
Table 2　Corresponding relationship between flight control system components and AADL components

| 飞控系统子系统或部件 | AADL 构件 |
|---|---|
| 飞控计算机 | System |
| 传感器 | System |
| IMU | System |
| 飞控计算机监控通道 | Process |
| 飞控计算机指令通道 | Process |
| 存储器 | Memory |
| 处理器 | Processor |
| 其他 | Abstract |

以该横侧向电传飞控系统的飞控计算机为例,说明系统组成部件的名义模型构建方法:飞控计算机输入信息包括飞行员控制信号、滚转角、滚转率以及偏航率等,其中控制信号为飞行员操纵事件,设置为事件端口,其余端口用于构件间信息交互,交互方式为传输数据,均设置为数据端口。飞控计算机的处理器和内存分别用处理器(Processor)构件和存储器(Memory)构件表示;每台飞控计算机有指令和监控两个通道,均用进程(Process)表示,与飞控计算机的硬件构件绑定,用于处理输入信息从而得到操纵面输出指令,但监控通道信息不作为输出,而是用于和指令通道输出信息进行比较以检测指令通道的正确输出,Set_signal 进程为虚拟进程,用于对指令通道和监控通道信息进行比较。建立 AADL 模型如图 2 所示。

```
system implementation PFC1.impl
Subcomponents
MEM: memory MEM;
CPU: processor CPU.impl;
Monitor1: process monitor1;
Command1: process command1;
Set_signal: process ;
properties
Actual_Processor_Binding => (reference(CPU.part1)) applies to Command1;
Actual_Processor_Binding => (reference(CPU.part2)) applies to Monitor1;
Actual_memory_Binding => (reference(MEM)) applies to Command;
Actual_memory_Binding => (reference(MEM)) applies to Monitor;
end PFC1.impl;
```

(a) 飞控计算机名义模型代码




(b) 飞控计算机名义模型

图 2　飞控计算机 AADL 名义模型

Fig. 2　The AADL nominal model of the flight control computer

其他部件名义模型的建模方法与飞控计算机相同。根据系统架构连接各部件间的数据端口可建立飞控系统名义模型。基于 AADL 建立的横侧向电传飞控系统名义模型如图 3 所示。横侧向电

传飞控系统采用双通道架构,由两台完全一样的主飞控计算机系统PFC1/2接收IMU信息和各操纵面传感器的操纵面位置信息,结合飞行员操纵杆和踏板的输入信号,为舵面驱动机构计算控制命令。每一个舵面由双冗余的舵面驱动机构驱动,如图RAA1和RAA2为一对舵面驱动机构,用于驱动右副翼(Right_aileron)。舵面驱动机构对中的每个构件分别与两个PFC连接,如图LAA1、RAA1和RA1接受PFC1指令,LAA2、RAA2和RA2接受PFC2指令,再根据指令驱动左副翼(Left_aileron)、右副翼(Right_aileron)和方向舵(Rudder)。

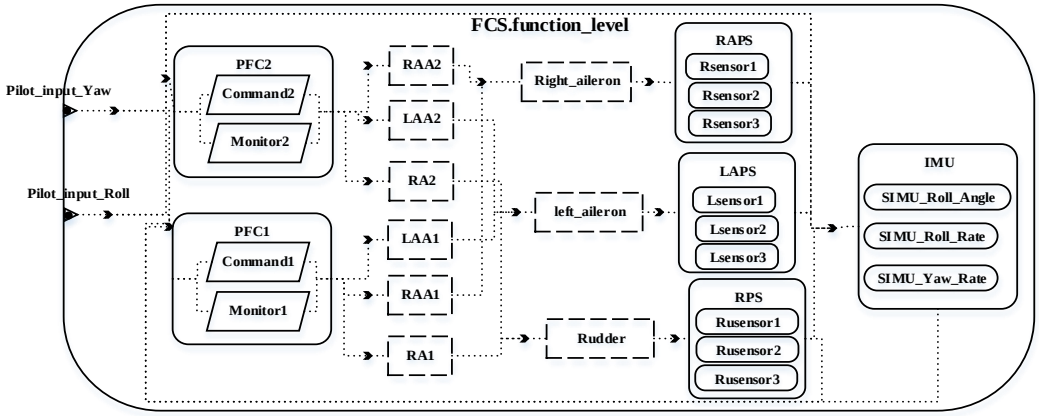



图3 飞控系统AADL名义模型

Fig. 3 The AADL nominal model of the flight control system

## 1.2 系统错误模型构建

AADL错误附件能对各构件的错误行为和构件间的错误传播建模。使用错误附件拓展名义模型时,首先建立错误附件库,声明各构件的错误类型和状态,再声明系统和各个构件的错误行为。

### 1.2.1 错误附件库

错误附件库包括了用于描述飞控系统各构件的错误状态、错误传播的错误类型和错误事件。

对飞控系统中构件可能发生的错误,本文定义了六种错误类型,各构件的错误事件及其失效率、错误状态以及传播错误类型如图4所示。以飞控计算机为例,Computer_Omission表示无输出信息的错误状态,该状态可以由错误事件E_Omission激活,当飞控计算机无响应时,会传出传播No_Response错误;Computer_Random表示随机输出信息的随机输出错误状态,该状态可以由错误事件E_Random激活,处于该状态时会传出传播Random_Output错误。

飞控计算机

| 错误事件 | 错误状态 | 传出传播错误 | 事件失效率 |
|---|---|---|---|
| E_Stuck | Computer_Stuck | Stuck_Fault | 1e-7 |
| E_Random | Computer_Random | Random_output | 1e-7 |
| E_Delayed | Computer_Delayed | Delayed_Fault | 1e-7 |
| E_Omission | Computer_Omission | No_Response | 2e-7 |

舵面驱动机构

| 错误事件 | 错误状态 | 传出传播错误 | 事件失效率 |
|---|---|---|---|
| E_Stuck | Stuck | Stuck_Fault | 1e-6 |
| E_Omission | Omission | No_Response | 1e-6 |

舵面

| 错误事件 | 错误状态 | 传出传播错误 | 事件失效率 |
|---|---|---|---|
| E_Stuck | Stuck | Stuck_Fault | 1e-8 |
| E_Trailing | Trailing | Singal_Excursion | 1e-8 |

传感器

| 错误事件 | 错误状态 | 传出传播错误 | 事件失效率 |
|---|---|---|---|
| E_Omission | Omission | No_response | 4e-7 |

IMU

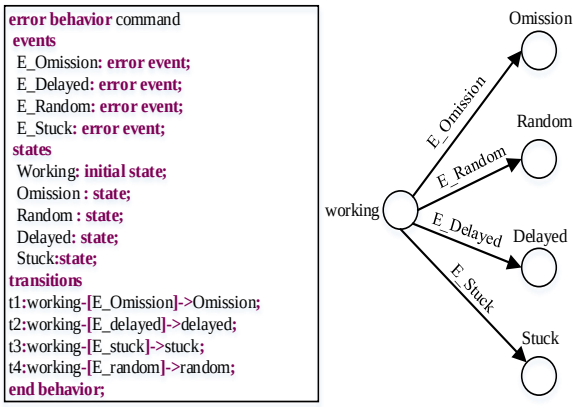
| 错误事件 | 错误状态 | 传出传播错误 | 事件失效率 |
|---|---|---|---|
| E_Omission | Omission | No_response | 4e-7 |
| E_Gain_change | Gain_change | Gain_variation | 3e-7 |

图4 各构件错误附件库信息

Fig. 4 The error library information of each component

在错误附件库中建立各构件的错误行为状态机,可以声明各构件的错误状态由于错误事件发生时的状态转换。以错误附件库中飞控计算机指令通道的错误行为状态机为例,如图5所示,定义

了四种用于激活指令通道状态转换的错误事件以及五种状态,指令通道的初始状态为working状态,表示正常工作状态,当发生错误事件如E_Delayed,指令通道将从working状态转换为Delay错误状态,如图5中t2所示。




(a) 指令通道错误行为状态机语义　　(b) 指令通道状态转移图

图5　指令通道行为状态机

Fig. 5　The error behavior states machine of the command lane

### 1.2.2　构件错误模型构建

构件错误模型包括三个部分:构件错误传播、构件错误行为和复合错误行为。本节将以一台飞控计算机的错误模型为例说明构件的错误行为建模方法。

1) 构件错误传播

构件错误传播可以指定构件间传入传播和传出传播的错误类型。传入传播错误可以声明构件接收的相连构件的传出传播错误。错误传出传播可以声明构件通过端口传出传播的错误类型。

飞控计算机错误模型如图6所示,飞控计算机通过端口PFC1_L_angle接收左副翼传感器数据,从而获取左副翼位置,所以当传感器处于错误状态时,传输错误左副翼位置数据,飞控计算机会由于通过端口PFC1_L_angle与之交互而接收到传感器传播的错误,如无响应错误(No_Response)。同理右副翼传感器和方向舵传感器也会传输错误右副翼和方向舵位置数据至飞控计算机。同样IMU系统也传输数据至飞控计算机,当IMU系统处于错误状态时也会传播错误至飞控计算机,包括无响应错误(No_Response)和输出增益变化错误(Gain_change)。飞控计算机处于错误状态时,也会通过数据输出端口PFC_signal_out传出传播错误,包括无响应错误(No_Response)和卡滞错误(Stuck_Fault)等。

2) 构件错误行为

构件错误行为描述了构件状态变换和处于某一状态时通过端口传出传播的错误类型。在图6的飞控计算机错误模型示例中,传感器和IMU会传出传播错误至飞控计算机,如图6中t1所示,指令单元接收到IMU传播的无响应错误或指令单元发生无响应错误事件时,将由工作状态转换为无响应状态。当飞控计算机处于错误状态时,会通过输出端口传出传播错误,影响其他交互构件的状态,如示例中p1所示,当飞控计算机处于无响应错误状态(Computer_Omission),会通过输出端口Signal_out传出传播No_response错误。结合错误传播行为和构件错误行为,可以描述飞控系统部件间的故障传播行为,声明错误传播中由于构件处于错误状态时传出传播的错误类型作用于其他构件,导致其他构件的状态变换。

3) 复合错误行为

复合错误行为可以定义系统中构件的失效模式到系统本身的错误状态的映射。在图6示例中,指令通道和监控通道作为飞控计算机的子构件,当指令通道和监控通道同时故障时,会导致飞控计算机故障,例如当指令通道和监控通道同时处于卡滞状态(Stuck)时,飞控计算机状态会变换至卡滞状态(Computer_Stuck)。

在错误模型中,可以在属性规约中声明错误事件发生或传播错误的函数分布,如指数分布和威布尔分布等。如图6所示,错误模型属性规约中声明了指令通道发生无响应错误事件和传入传播传感器错误发生的概率分布类型,且均为指数分布。对于飞控系统模型中其他构件如传感器、舵面驱动机构以及操纵面等,通过相同方式利用构件错误传播、构件错误行为和复合错误行为构建构件错误模型,从而建立横侧向电传飞控系统完整错误模型。

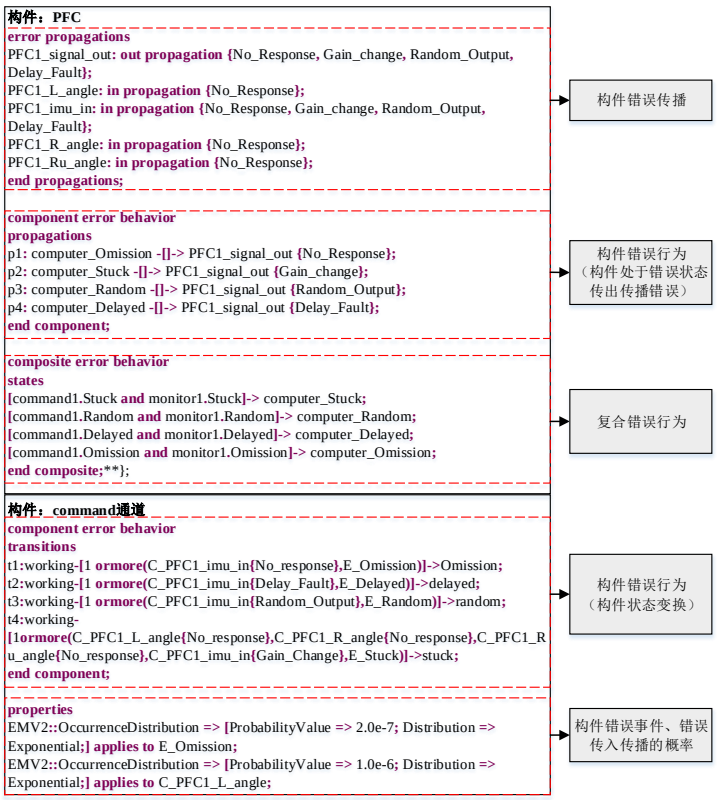



图 6 飞控计算机错误模型

Fig. 6 The error model of the flight control system

## 2 基于 AADL 模型的 SPN 构建

随机 Petri 网定义为一个七元组，即 $\sum=(P, T, F, K, W, M_0, \Lambda)$，其中：

1) $P=\{p_1, p_2, \cdots, p_n\}$，表示库所的集合。

2) $T=\{t_1, t_2, \cdots, t_m\}$，表示变迁的集合。

3) $F \subseteq (P \times T) \cup (T \times P)$，表示连接库所和变迁的弧的集合。

4) $K: P \rightarrow \{1, 2, 3, \cdots\}$ 为库所的容量函数。

5) $W: F \rightarrow \{1, 2, 3, \cdots\}$ 为弧的权重函数。

6) $M: P \rightarrow \{0, 1, 2, \cdots\}$ 表示网络中库所的标识，且 $\forall p \in P: M(p) \leqslant K(p)$，$M_0$ 表示初始网络的标识。

7) $\Lambda=\{\lambda_1, \lambda_2, \cdots, \lambda_m\}$，为变迁的触发速率，其中，$\lambda_i (i=1, 2, \cdots m)$，表示第 $i$ 个变迁被触发的速率。

本节将从飞控系统 AADL 模型中提取用于建立 SPN 模型的信息，AADL 模型元素与 SPN 模型元素的映射规则如下。元素映射关系如表 3 所示。

表 3 AADL 与 SPN 元素映射关系

Table 3 The mapping relationship between AADL element and SPN element

| | AADL 模型 | SPN 模型 |
|---|---|---|
| 表达式 | 状态－[转移条件(错误事件或传播错误、转换逻辑)]－>状态 | 库所－弧－库所 |
| 元素 | 初始状态 | 初始标识为 1 的库所 |
| | 非初始状态 | 初始标识为 0 的库所 |
| | 错误事件、传播错误 | 变迁 |
| | 错误事件、传播错误的失效率 | 变迁的触发速率 |
| | 状态到转换条件 | 相应库所到相应变迁的弧 |
| | 转换条件到状态 | 相应变迁到相应库所的弧 |

规则 1：AADL 模型中构件的状态映射为 SPN 模型的库所，其中初始状态对应库所的初始标识为 1，其他库所的初始标识为 0。

规则 2：库所的容量与构件的状态转移条件中转移逻辑相关。如状态转移条件为 2/3 表决逻辑，则转换到的状态对应库所的容量为 2，若为与逻辑、或逻辑，则转换到的状态对应的库所容量为 1。

规则 3：AADL 模型中构件的状态转移条件映

射为 SPN 模型的变迁,错误事件或传播错误的失效率映射为变迁触发速率。

规则 4:AADL 模型中构件的状态转移过程映射为 SPN 的弧。转换前状态到转换条件映射为相应库所连接相应变迁的弧,转换条件到转换后状态映射为相应变迁连接相应库所的弧。弧的权重与库所容量相关,传出传播错误的状态对应库所与传播错误对应变迁之间弧的权重等于该库所的容量。

## 2.1　AADL 模型库所信息提取

与库所对应的 AADL 模型元素为构件状态,从构件错误模型中提取构件错误行为和复合错误行为中状态变换的前后状态确定库所。对于构件的初始状态,提取到其对应的库所命名为构件的名称,初始标识数为 1,对于其他错误状态,提取到其对应的库所命名为“构件名称/构件状态”,初始标识数为 0。库所的容量与构件的状态转移条件相关,如构件状态转移为 2/3 表决逻辑,转移条件为发生两个错误事件或同时接收到两个构件的传入传播错误,则构件的错误状态对应库所的容量为 2。该飞控系统模型中,除传感器和 IMU 错误状态对应的库所容量为 2 外,其余库所容量均为 1。

左副翼构件的库所信息提取过程如图 7 所示。左副翼的初始状态对应的库所命名为 Left_aileron,初始标识数为 1。左副翼构件会由于错误事件或传入传播错误由初始状态转换为左副翼卡滞状态或左副翼漂移状态,两个错误状态对应的库所名称分别为 Left_aileron/Stuck 和 Left_aileron/trailing,初始标识数均为 0。飞控系统提取到 66 个库所,如表 4 所示。

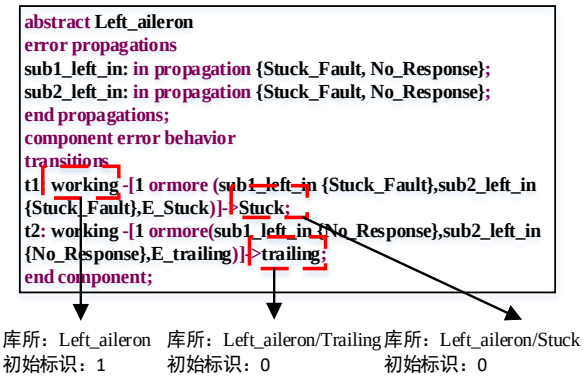



图 7　左副翼库所信息提取

Fig. 7　Information of places related to the left aileron

表 4　飞控系统提取到库所

Table 4　The places extracted from the flight control system

| 序号 | 库所名 | 序号 | 库所名 | 序号 | 库所名 | 序号 | 库所名 | 序号 | 库所名 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | IMU_Roll_Angle | 2 | IMU_Roll_Rate | 3 | IMU_Yaw_Rate | 4 | lsensor_3 | 5 | lsensor_2 |
| 6 | lsensor_1 | 7 | rsensor_3 | 8 | rsensor_2 | 9 | rsensor_1 | 10 | rusensor_3 |
| 11 | rusensor_2 | 12 | rusensor_1 | 13 | LAPS/Omission | 14 | RAPS/Omission | 15 | RPS/Omission |
| 16 | IMU/Omission | 17 | IMU/Gain_Change | 18 | IMU/Rao_yg | 19 | IMU/Rao_rg | 20 | monitor1 |
| 21 | monitor1/Omission | 22 | monitor1/delayed | 23 | monitor1/random | 24 | monitor1/stuck | 25 | command1 |
| 26 | command1/Omission | 27 | command1/delayed | 28 | command1/random | 29 | command1/stuck | 30 | monitor2 |
| 31 | monitor2/Omission | 32 | monitor2/delayed | 33 | monitor2/random | 34 | monitor2/stuck | 35 | command2 |
| 36 | command2/Omission | 37 | command2/delayed | 38 | command2/random | 39 | command2/stuck | 40 | RA1 |
| 41 | RA1/Omission | 42 | RA1/Stuck | 43 | RAA1 | 44 | RAA1/Omission | 45 | RAA1/Stuck |
| 46 | LAA1 | 47 | LAA1/Omission | 48 | LAA1/Stuck | 49 | RA2 | 50 | RA2/Omission |
| 51 | RA2/Stuck | 52 | RAA2 | 53 | RAA2/Omission | 54 | RAA2/Stuck | 55 | LAA2 |
| 56 | LAA2/Omission | 57 | LAA2/Stuck | 58 | rudder | 59 | rudder/Stuck | 60 | rudder/trailing |
| 61 | Right_aileron | 62 | Right_aileron/Stuck | 63 | Right_aileron/trailing | 64 | Left_aileron | 65 | Left_aileron/Stuck |
| 66 | Left_aileron/trailing | | | | | | | | |

## 2.2　AADL 模型变迁信息提取

变迁是导致库所中标识转移的条件,对应于构件中状态变换的条件,即错误事件和传播错误。对于错误事件,一个错误事件对应一个变迁,命名

为“构件名称/错误事件名称”。对于错误传播，一个构件传播至另一个构件的错误对应一个变迁，命名为“传播构件_传出错误的构件状态*传入传播构件/p”。变迁的触发速率从构件错误模型中对错误事件和传播错误的属性规约中提取。

左副翼变迁的提取过程如图8所示。左副翼状态会由于一个错误事件和来自两个舵面驱动机构的传入传播错误变换状态，由初始状态变换为卡滞状态，因此提取到三个变迁，对应于左副翼构件的错误事件和两个传入传播错误，各变迁的触发速率可通过属性规约获得。飞控系统模型中共提取到115个变迁，如表5所示。

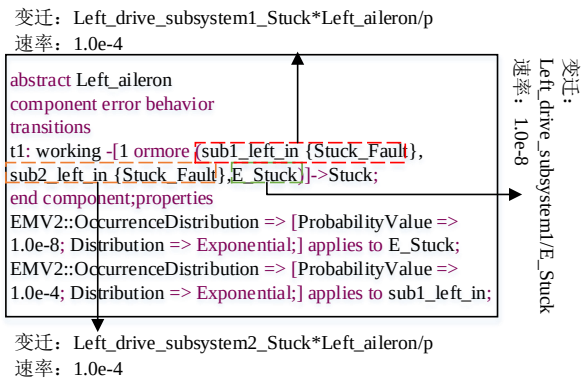



图8 左副翼变迁信息提取

Fig. 8 Information of transitions related to the left aileron

表5 飞控系统提取到变迁

Table 5 The transitions extracted from the flight control system

| 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | IMU_Roll_Angle/E_Omission | 2 | IMU_Roll_Angle/E_Gain_Change | 3 | IMU_Roll_Rate/E_Omission | 4 | IMU_Roll_Rate/E_Gain_Change | 5 | IMU_Yaw_Rate/E_Omission |
| 6 | IMU_Yaw_Rate/E_Gain_Change | 7 | lsensor_3/E_Omission | 8 | lsensor_2/E_Omission | 9 | lsensor_1/E_Omission | 10 | rsensor_3/E_Omission |
| 11 | rsensor_2/E_Omission | 12 | rsensor_1/E_Omission | 13 | rusensor_3/E_Omission | 14 | rusensor_2/E_Omission | 15 | rusensor_1/E_Omission |
| 16 | IMU/Rao_yg*and | 17 | IMU/Rao_rg*and | 18 | IMU_Omission*monitor1/p | 19 | monitor1/E_Omission | 20 | IMU_Rao_rg*monitor1/p |
| 21 | monitor1/E_Delayed | 22 | IMU_Rao_yg*monitor1/p | 23 | monitor1/E_Random | 24 | LAPS_Omission*monitor1/p | 25 | RAPS_Omission*monitor1/p |
| 26 | RPS_Omission*monitor1/p | 27 | IMU_Gain_Change*monitor1/p | 28 | monitor1/E_Stuck | 29 | IMU_Omission*command1/p | 30 | command1/E_Omission |
| 31 | IMU_Rao_rg*command1/p | 32 | command1/E_Delayed | 33 | IMU_Rao_yg*command1/p | 34 | command1/E_Random | 35 | LAPS_Omission*command1/p |
| 36 | RAPS_Omission*command1/p | 37 | RPS_Omission*command1/p | 38 | IMU_Gain_Change*command1/p | 39 | command1/E_Stuck | 40 | IMU_Omission*monitor2/p |
| 41 | monitor2/E_Omission | 42 | IMU_Rao_rg*monitor2/p | 43 | monitor2/E_Delayed | 44 | IMU_Rao_yg*monitor2/p | 45 | monitor2/E_Random |
| 46 | LAPS_Omission*monitor2/p | 47 | RAPS_Omission*monitor2/p | 48 | RPS_Omission*monitor2/p | 49 | IMU_Gain_Change*monitor2/p | 50 | monitor2/E_Stuck |
| 51 | IMU_Omission*command2/p | 52 | command2/E_Omission | 53 | IMU_Rao_rg*command2/p | 54 | command2/E_Delayed | 55 | IMU_Rao_yg*command2/p |
| 56 | command2/E_Random | 57 | LAPS_Omission*command2/p | 58 | RAPS_Omission*command2/p | 59 | RPS_Omission*command2/p | 60 | IMU_Gain_Change*command2/p |
| 61 | command2/E_Stuck | 62 | PFC1_computer_Omission*RA1/p | 63 | RA1/E_Omission | 64 | PFC1_computer_Random*RA1/p | 65 | PFC1_computer_Stuck*RA1/p |
| 66 | PFC1_computer_Delayed*RA1/p | 67 | RA1/E_stuck | 68 | PFC1_computer_Omission*RAA1/p | 69 | RAA1/E_Omission | 70 | PFC1_computer_Random*RAA1/p |

续表

| 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 | 序号 | 变迁名 |
|---|---|---|---|---|---|---|---|---|---|
| 71 | PFC1_computer_Stuck*RAA1/p | 72 | PFC1_computer_Delayed*RAA1/p | 73 | RAA1/E_stuck | 74 | PFC1_computer_Omission*LAA1/p | 75 | LAA1/E_Omission |
| 76 | PFC1_computer_Random*LAA1/p | 77 | PFC1_computer_Stuck*LAA1/p | 78 | PFC1_computer_Delayed*LAA1/p | 79 | LAA1/E_stuck | 80 | PFC2_computer_Omission*RA2/p |
| 81 | RA2/E_Omission | 82 | PFC2_computer_Random*RA2/p | 83 | PFC2_computer_Stuck*RA2/p | 84 | PFC2_computer_Delayed*RA2/p | 85 | RA2/E_stuck |
| 86 | PFC2_computer_Omission*RAA2/p | 87 | RAA2/E_Omission | 88 | PFC2_computer_Random*RAA2/p | 89 | PFC2_computer_Stuck*RAA2/p | 90 | PFC2_computer_Delayed*RAA2/p |
| 91 | RAA2/E_stuck | 92 | PFC2_computer_Omission*LAA2/p | 93 | LAA2/E_Omission | 94 | PFC2_computer_Random*LAA2/p | 95 | PFC2_computer_Stuck*LAA2/p |
| 96 | PFC2_computer_Delayed*LAA2/p | 97 | LAA2/E_stuck | 98 | RA1_Stuck*rudder/p | 99 | RA2_Stuck*rudder/p | 100 | rudder/E_Stuck |
| 101 | RA1_Omission*rudder/p | 102 | RA2_Omission*rudder/p | 103 | rudder/E_trailing | 104 | RAA1_Stuck*Right_aileron/p | 105 | RAA2_Stuck*Right_aileron/p |
| 106 | Right_aileron/E_Stuck | 107 | RAA1_Omission*Right_aileron/p | 108 | RAA2_Omission*Right_aileron/p | 109 | Right_aileron/E_trailing | 110 | LAA1_Stuck*Left_aileron/p |
| 111 | LAA2_Stuck*Left_aileron/p | 112 | Left_aileron/E_Stuck | 113 | LAA1_Omission*Left_aileron/p | 114 | LAA2_Omission*Left_aileron/p | 115 | Left_aileron/E_trailing |

## 2.3　AADL模型弧信息提取

对于由错误事件导致的状态变换,对应于连接构件状态变换前状态对应的库所、状态变换后库所、错误事件对应的变迁和两条弧构成的子网。连接情况为:一条弧由变换前状态对应的库所连接至错误事件对应的变迁,另一条弧由变迁连接至状态变换后状态对应的库所。该类状态变换中各条弧的权值均为1。

对于传播错误导致的状态变换,对应于传播错误对应的变迁、传出传播该错误的构件状态对应的库所、传入传播该错误的构件的变换前状态对应的库所、变换后状态对应的库所和四条弧构成的子网。连接情况为:一条弧由传出传播该错误的构件状态对应的库所连接至传播错误对应的变迁,一条弧由变迁连接至传出传播该错误的构件状态对应的库所,一条弧由传入传播该错误的构件的变换前状态对应的库所连接至变迁,一条弧由变迁连接至传入传播该错误的构件的变换后状态对应的库所。该类状态变换中弧的权值与库所的标识数有关,例如传感器无响应状态库所、IMU无响应和增益变化状态库所容量为2,当其故障影响其他构件时,该库所须具有两个标识,当其影响其他构件时,连接传出传播错误对应变迁的弧和该变迁连接至该构件错误状态对应库所的弧的权值均为2,其他弧权值为1。

对于飞控计算机和传感器系统等具有复合错误行为的构件,若错误状态对应库所的容量大于1,如传感器系统,连接情况为:各传感器初始状态对应的库所连接至各传感器错误事件对应的变迁,变迁连接至系统错误状态对应的库所,各弧的权值均为1。若系统错误状态的容量为1,如飞控计算机,传入传播错误影响的构件为两个通道,传出传播错误由两个通道传出,例如飞控计算机传出传播卡滞错误表现为两个通道同时传出传播卡滞错误,即两条弧由两个通道卡滞状态对应的库所连接至传播卡滞错误对应的变迁,各弧的权值均为1。

从飞控系统模型提取到表示库所到变迁的连接矩阵 $W_{pt}$ 中非 0 元素如图 9 所示，表示变迁到库所的连接矩阵 $W_{tp}$ 中非 0 元素如图 10 所示，如图 $w_{pt}(i,j)=k$ 或 $w_{tp}(i,j)=k$ 中 $i$、$j$ 分别表示库所和变迁的序号，$k$ 表示该条弧的权值。例如存在权值为 1 的弧由序号为 64 的库所（左副翼初始状态）连接至序号为 112 的变迁（左副翼卡滞事件），则 $w_{pt}(64,112)=1$。例如存在权值为 1 的弧由序号为 112 的变迁（左副翼卡滞事件）连接至序号为 65 的库所（左副翼卡滞状态），则 $w_{tp}(65,112)=1$。

$$
\begin{gathered}
w_{pt}(1,1)=1, w_{pt}(1,2)=1, w_{pt}(1,16)=1, w_{pt}(1,17)=1, w_{pt}(2,3)=1,\\
w_{pt}(2,4)=1, w_{pt}(2,17)=1, w_{pt}(3,5)=1, w_{pt}(3,6)=1, w_{pt}(3,16)=1,\\
w_{pt}(4,7)=1, w_{pt}(5,8)=1, w_{pt}(6,9)=1, w_{pt}(7,10)=1, w_{pt}(8,11)=1,\\
w_{pt}(9,12)=1, w_{pt}(10,13)=1, w_{pt}(11,14)=1, w_{pt}(12,15)=1, w_{pt}(13,24)=2,\\
w_{pt}(13,35)=2, w_{pt}(13,46)=2, w_{pt}(13,57)=2, w_{pt}(14,25)=2, w_{pt}(14,36)=2,\\
w_{pt}(14,47)=2, w_{pt}(14,58)=2, w_{pt}(15,26)=2, w_{pt}(15,37)=2, w_{pt}(15,48)=2,\\
w_{pt}(15,59)=2, w_{pt}(16,18)=2, w_{pt}(16,29)=2, w_{pt}(16,40)=2, w_{pt}(16,51)=2,\\
w_{pt}(17,27)=2, w_{pt}(17,38)=2, w_{pt}(17,49)=2, w_{pt}(17,60)=2, w_{pt}(18,22)=1,\\
w_{pt}(18,33)=1, w_{pt}(18,44)=1, w_{pt}(18,55)=1, w_{pt}(19,20)=1, w_{pt}(19,31)=1,\\
w_{pt}(19,42)=1, w_{pt}(19,53)=1, w_{pt}(20,18)=1, w_{pt}(20,19)=1, w_{pt}(20,20)=1,\\
w_{pt}(20,21)=1, w_{pt}(20,22)=1, w_{pt}(20,23)=1, w_{pt}(20,24)=1, w_{pt}(20,25)=1,\\
w_{pt}(20,26)=1, w_{pt}(20,27)=1, w_{pt}(20,28)=1, w_{pt}(21,62)=1, w_{pt}(21,68)=1,\\
w_{pt}(21,74)=1, w_{pt}(22,66)=1, w_{pt}(22,72)=1, w_{pt}(22,78)=1, w_{pt}(23,64)=1,\\
w_{pt}(23,70)=1, w_{pt}(23,76)=1, w_{pt}(24,65)=1, w_{pt}(24,71)=1, w_{pt}(24,77)=1,\\
w_{pt}(25,29)=1, w_{pt}(25,30)=1, w_{pt}(25,31)=1, w_{pt}(25,32)=1, w_{pt}(25,33)=1,\\
w_{pt}(25,34)=1, w_{pt}(25,35)=1, w_{pt}(25,36)=1, w_{pt}(25,37)=1, w_{pt}(25,38)=1,\\
w_{pt}(25,39)=1, w_{pt}(26,62)=1, w_{pt}(26,68)=1, w_{pt}(26,74)=1, w_{pt}(27,66)=1,\\
w_{pt}(27,72)=1, w_{pt}(27,78)=1, w_{pt}(28,64)=1, w_{pt}(28,70)=1, w_{pt}(28,76)=1,\\
w_{pt}(29,65)=1, w_{pt}(29,71)=1, w_{pt}(29,77)=1, w_{pt}(30,40)=1, w_{pt}(30,41)=1,\\
w_{pt}(30,42)=1, w_{pt}(30,43)=1, w_{pt}(30,44)=1, w_{pt}(30,45)=1, w_{pt}(30,46)=1,\\
w_{pt}(30,47)=1, w_{pt}(30,48)=1, w_{pt}(30,49)=1, w_{pt}(30,50)=1, w_{pt}(31,80)=1,\\
w_{pt}(31,86)=1, w_{pt}(31,92)=1, w_{pt}(32,84)=1, w_{pt}(32,90)=1, w_{pt}(32,96)=1,\\
w_{pt}(33,82)=1, w_{pt}(33,88)=1, w_{pt}(33,94)=1, w_{pt}(34,83)=1, w_{pt}(34,89)=1,\\
w_{pt}(34,95)=1, w_{pt}(35,51)=1, w_{pt}(35,52)=1, w_{pt}(35,53)=1, w_{pt}(35,54)=1,\\
w_{pt}(35,55)=1, w_{pt}(35,56)=1, w_{pt}(35,57)=1, w_{pt}(35,58)=1, w_{pt}(35,59)=1,\\
w_{pt}(35,60)=1, w_{pt}(35,61)=1, w_{pt}(36,80)=1, w_{pt}(36,86)=1, w_{pt}(36,92)=1,\\
w_{pt}(37,84)=1, w_{pt}(37,90)=1, w_{pt}(37,96)=1, w_{pt}(38,82)=1, w_{pt}(38,88)=1,\\
w_{pt}(38,94)=1, w_{pt}(39,83)=1, w_{pt}(39,89)=1, w_{pt}(39,95)=1, w_{pt}(40,62)=1,\\
w_{pt}(40,63)=1, w_{pt}(40,64)=1, w_{pt}(40,65)=1, w_{pt}(40,66)=1, w_{pt}(40,67)=1,\\
w_{pt}(41,101)=1, w_{pt}(42,98)=1, w_{pt}(43,68)=1, w_{pt}(43,69)=1, w_{pt}(43,70)=1,\\
w_{pt}(43,71)=1, w_{pt}(43,72)=1, w_{pt}(43,73)=1, w_{pt}(44,107)=1, w_{pt}(45,104)=1,\\
w_{pt}(46,74)=1, w_{pt}(46,75)=1, w_{pt}(46,76)=1, w_{pt}(46,77)=1, w_{pt}(46,78)=1,\\
w_{pt}(46,79)=1, w_{pt}(47,113)=1, w_{pt}(48,110)=1, w_{pt}(49,80)=1, w_{pt}(49,81)=1,\\
w_{pt}(49,82)=1, w_{pt}(49,83)=1, w_{pt}(49,84)=1, w_{pt}(49,85)=1, w_{pt}(50,102)=1,\\
w_{pt}(51,99)=1, w_{pt}(52,86)=1, w_{pt}(52,87)=1, w_{pt}(52,88)=1, w_{pt}(52,89)=1,\\
w_{pt}(52,90)=1, w_{pt}(52,91)=1, w_{pt}(53,108)=1, w_{pt}(54,105)=1, w_{pt}(55,92)=1,\\
w_{pt}(55,93)=1, w_{pt}(55,94)=1, w_{pt}(55,95)=1, w_{pt}(55,96)=1, w_{pt}(55,97)=1,\\
w_{pt}(56,114)=1, w_{pt}(57,111)=1, w_{pt}(58,98)=1, w_{pt}(58,99)=1, w_{pt}(58,100)=1,\\
w_{pt}(58,101)=1, w_{pt}(58,102)=1, w_{pt}(58,103)=1, w_{pt}(61,104)=1, w_{pt}(61,105)=1,\\
w_{pt}(61,106)=1, w_{pt}(61,107)=1, w_{pt}(61,108)=1, w_{pt}(61,109)=1, w_{pt}(64,110)=1,\\
w_{pt}(64,111)=1, w_{pt}(64,112)=1, w_{pt}(64,113)=1, w_{pt}(64,114)=1, w_{pt}(64,115)=1
\end{gathered}
$$

图 9 $W_{pt}$ 矩阵中非 0 元素

Fig. 9 The non zero elements in a matrix $W_{pt}$

$$
\begin{gathered}
w_{tp}(13,7)=1, w_{tp}(13,8)=1, w_{tp}(13,9)=1, w_{tp}(13,24)=2, w_{tp}(13,35)=2,\\
w_{tp}(13,46)=2, w_{tp}(13,57)=2, w_{tp}(14,10)=1, w_{tp}(14,11)=1, w_{tp}(14,12)=1,\\
w_{tp}(14,25)=2, w_{tp}(14,36)=2, w_{tp}(14,47)=2, w_{tp}(14,58)=2, w_{tp}(15,13)=1,\\
w_{tp}(15,14)=1, w_{tp}(15,15)=1, w_{tp}(15,26)=2, w_{tp}(15,37)=2, w_{tp}(15,48)=2,\\
w_{tp}(15,59)=2, w_{tp}(16,1)=1, w_{tp}(16,3)=1, w_{tp}(16,5)=1, w_{tp}(16,18)=2,\\
w_{tp}(16,29)=2, w_{tp}(16,40)=2, w_{tp}(16,51)=2, w_{tp}(17,2)=1, w_{tp}(17,4)=1,\\
w_{tp}(17,6)=1, w_{tp}(17,27)=2, w_{tp}(17,38)=2, w_{tp}(17,49)=2, w_{tp}(17,60)=2,\\
w_{tp}(18,16)=1, w_{tp}(18,22)=1, w_{tp}(18,33)=1, w_{tp}(18,44)=1, w_{tp}(18,55)=1,\\
w_{tp}(19,17)=1, w_{tp}(19,20)=1, w_{tp}(19,31)=1, w_{tp}(19,42)=1, w_{tp}(19,53)=1,\\
w_{tp}(21,18)=1, w_{tp}(21,19)=1, w_{tp}(21,62)=1, w_{tp}(21,68)=1, w_{tp}(21,74)=1,\\
w_{tp}(22,20)=1, w_{tp}(22,21)=1, w_{tp}(22,66)=1, w_{tp}(22,72)=1, w_{tp}(22,78)=1,\\
w_{tp}(23,22)=1, w_{tp}(23,23)=1, w_{tp}(23,64)=1, w_{tp}(23,70)=1, w_{tp}(23,76)=1,\\
w_{tp}(24,24)=1, w_{tp}(24,25)=1, w_{tp}(24,26)=1, w_{tp}(24,27)=1, w_{tp}(24,28)=1,\\
w_{tp}(24,65)=1, w_{tp}(24,71)=1, w_{tp}(24,77)=1, w_{tp}(26,29)=1, w_{tp}(26,30)=1,\\
w_{tp}(26,62)=1, w_{tp}(26,68)=1, w_{tp}(26,74)=1, w_{tp}(27,31)=1, w_{tp}(27,32)=1,\\
w_{tp}(27,66)=1, w_{tp}(27,72)=1, w_{tp}(27,78)=1, w_{tp}(28,33)=1, w_{tp}(28,34)=1,\\
w_{tp}(28,64)=1, w_{tp}(28,70)=1, w_{tp}(28,76)=1, w_{tp}(29,35)=1, w_{tp}(29,36)=1,\\
w_{tp}(29,37)=1, w_{tp}(29,38)=1, w_{tp}(29,39)=1, w_{tp}(29,65)=1, w_{tp}(29,71)=1,\\
w_{tp}(29,77)=1, w_{tp}(31,40)=1, w_{tp}(31,41)=1, w_{tp}(31,80)=1, w_{tp}(31,86)=1,\\
w_{tp}(31,92)=1, w_{tp}(32,42)=1, w_{tp}(32,43)=1, w_{tp}(32,84)=1, w_{tp}(32,90)=1,\\
w_{tp}(32,96)=1, w_{tp}(33,44)=1, w_{tp}(33,45)=1, w_{tp}(33,82)=1, w_{tp}(33,88)=1,\\
w_{tp}(33,94)=1, w_{tp}(34,46)=1, w_{tp}(34,47)=1, w_{tp}(34,48)=1, w_{tp}(34,49)=1,\\
w_{tp}(34,50)=1, w_{tp}(34,83)=1, w_{tp}(34,89)=1, w_{tp}(34,95)=1, w_{tp}(36,51)=1,\\
w_{tp}(36,52)=1, w_{tp}(36,80)=1, w_{tp}(36,86)=1, w_{tp}(36,92)=1, w_{tp}(37,53)=1,\\
w_{tp}(37,54)=1, w_{tp}(37,84)=1, w_{tp}(37,90)=1, w_{tp}(37,96)=1, w_{tp}(38,55)=1,\\
w_{tp}(38,56)=1, w_{tp}(38,82)=1, w_{tp}(38,88)=1, w_{tp}(38,94)=1, w_{tp}(39,57)=1,\\
w_{tp}(39,58)=1, w_{tp}(39,59)=1, w_{tp}(39,60)=1, w_{tp}(39,61)=1, w_{tp}(39,83)=1,\\
w_{tp}(39,89)=1, w_{tp}(39,95)=1, w_{tp}(41,62)=1, w_{tp}(41,63)=1, w_{tp}(41,101)=1,\\
w_{tp}(42,64)=1, w_{tp}(42,65)=1, w_{tp}(42,66)=1, w_{tp}(42,67)=1, w_{tp}(42,98)=1,\\
w_{tp}(44,68)=1, w_{tp}(44,69)=1, w_{tp}(44,107)=1, w_{tp}(45,70)=1, w_{tp}(45,71)=1,\\
w_{tp}(45,72)=1, w_{tp}(45,73)=1, w_{tp}(45,104)=1, w_{tp}(47,74)=1, w_{tp}(47,75)=1,\\
w_{tp}(47,113)=1, w_{tp}(48,76)=1, w_{tp}(48,77)=1, w_{tp}(48,78)=1, w_{tp}(48,79)=1,\\
w_{tp}(48,110)=1, w_{tp}(50,80)=1, w_{tp}(50,81)=1, w_{tp}(50,102)=1, w_{tp}(51,82)=1,\\
w_{tp}(51,83)=1, w_{tp}(51,84)=1, w_{tp}(51,85)=1, w_{tp}(51,99)=1, w_{tp}(53,86)=1,\\
w_{tp}(53,87)=1, w_{tp}(53,108)=1, w_{tp}(54,88)=1, w_{tp}(54,89)=1, w_{tp}(54,90)=1,\\
w_{tp}(54,91)=1, w_{tp}(54,105)=1, w_{tp}(56,92)=1, w_{tp}(56,93)=1, w_{tp}(56,114)=1,\\
w_{tp}(57,94)=1, w_{tp}(57,95)=1, w_{tp}(57,96)=1, w_{tp}(57,97)=1, w_{tp}(57,111)=1,\\
w_{tp}(59,98)=1, w_{tp}(59,99)=1, w_{tp}(59,100)=1, w_{tp}(60,101)=1, w_{tp}(60,102)=1,\\
w_{tp}(60,103)=1, w_{tp}(62,104)=1, w_{tp}(62,105)=1, w_{tp}(62,106)=1, w_{tp}(63,107)=1,\\
w_{tp}(63,108)=1, w_{tp}(63,109)=1, w_{tp}(65,110)=1, w_{tp}(65,111)=1, w_{tp}(65,112)=1,\\
w_{tp}(66,113)=1, w_{tp}(66,114)=1, w_{tp}(66,115)=1
\end{gathered}
$$

图 10 $W_{tp}$ 矩阵中的非 0 元素

Fig. 10 The non zero elements in a matrix $W_{tp}$

## 3 基于蒙特卡洛的系统可靠性评估

本节将基于SPN利用蒙特卡洛仿真方法生成系统的寿命样本，利用寿命样本计算系统的可靠性参数。

蒙特卡洛的仿真过程所需的输入参数包括：库所与变迁连接矩阵 $W_{pt}$、变迁与库所连接矩阵 $W_{tp}$、初始库所标识矩阵 $M_0$、库所容量函数 $K(\bullet)$、变迁的触发率矩阵 $\boldsymbol{\Lambda}=[\lambda_1,\lambda_2,\cdots,\lambda_m]$ 和蒙特卡洛最大仿真次数。仿真过程如图 11 所示。具体仿真步骤如下：

步骤 1：变量初始化。初始化当前的库所标识、当前时间和变迁的触发时间，即令

$M_{current}=M_0$、$\pi_{current}=0$和$\pi_j=0$,其中$M_{current}$表示当前时刻SPN的库所标识情况,$\pi_{current}$表示当前时刻,$\pi_j$表示第$j$个变迁的触发时间,$j=1,2,\cdots,m$。

步骤2:确定触发变迁。判断各个变迁能否触发,触发规则为:

$$\begin{cases}\forall p\in \bullet t: M(p)\geqslant W(p,t)\\ \forall p\in t\bullet-\bullet t: M(p)+W(t,p)\leqslant K(p)\\ \forall p\in t\bullet\cap\bullet t: M(p)+W(t,p)-W(p,t)\leqslant K(p)\end{cases}\tag{1}$$

式中:$\bullet t$和$t\bullet$分别为变迁t的前置集和后置集。

$$\begin{aligned}\bullet t&=\{y|(y\in P\cup T)\cap((y,t)\in F)\}\\ t\bullet&=\{y|(y\in P\cup T)\cap((t,y)\in F)\}\end{aligned}\tag{2}$$

更新每一个变迁的触发时间,若变迁不能触发,则其触发时间为0;若能触发,则根据其触发速率生成随机数;若多个变迁均能触发,则从所有能触发的变迁中选取触发时间最小的变迁触发,若多个变迁的触发时间相同且都为最小触发时间,则随机选择一个变迁触发[21]。随后根据$\pi_{current}=\pi_{current}+\pi_{\min}$更新当前时间,其中$\pi_{\min}$为最小触发时间。再对所有变迁的触发事件更新,更新方法:$\pi_j=\pi_j-\pi_{\min}$。

步骤3:更新库所标识。根据变迁触发情况确定当前的库所标识,库所更新方法为

$$\forall p\in P: M'(p)=\begin{cases}M(p)-W(p,t), p\in\bullet t-t\bullet\\ M(p)+W(p,t), p\in t\bullet-\bullet t\\ M(p)+W(t,p)-W(p,t), p\in\bullet t\cap t\bullet\\ M(p), otherwise\end{cases}\tag{3}$$

步骤4:判断一次仿真是否停止。判断此时的目标库所的标识与该库所的容量函数是否相同,即判断$M_{current}$和$K(\bullet)$中对应目标库所的元素是否相等,$M_{current}$不等于$K(\bullet)$,则返回步骤2继续仿真过程。若相等则终止当前模拟,当前的时间$\pi_{current}$即为系统的一个寿命样本。再判断是否达到最大仿真次数,若达到则最大仿真次数结束仿真过程,若未达到则仿真次数加一并返回步骤1开始新一次仿真。

利用提取得到SPN模型应用蒙特卡洛仿真过程,例如分析电传飞控系统左副翼漂移故障时,通过蒙特卡洛方法获取系统SPN模型中左副翼漂移故障对应库所的标识由初始标识变为目标容量的时刻,即获取了一个左副翼漂移故障的寿命样本数据,重复仿真过程10 000次,即可得到10 000个左副翼漂移故障的寿命样本。

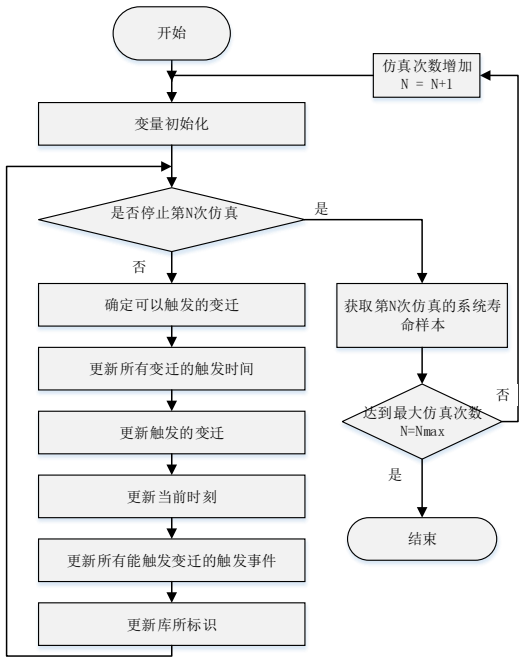



图11　蒙特卡洛仿真过程

Fig. 11　The process of Monte Carlo simulation

由于威布尔分布可以描述失效率递增、恒定或递减等多种情况的寿命分布类型,是可靠性中常用的失效分布类型,本节将利用威布尔分布拟合寿命样本,其表达式为

$$\begin{aligned}P(t)&=e^{-\left(\frac{t}{\eta}\right)^m}\\ \ln\left(\ln\left(\frac{1}{P(t)}\right)\right)&=m\ln(t)-m\ln(\eta)\end{aligned}\tag{4}$$

式中:$P(t)$为系统不处于指定错误状态的概率;$m$为形状参数;$\eta$为尺度参数。

根据仿真所得寿命样本,在$t_i(i=1,2,\cdots,N_{\min})$时刻的概率估计值为$\hat{P}=(N_{\max}-i)/N_{\max}$,其中$N_{\max}$为寿命样本个数。通过极大似然估计可确定威布尔分布中的形状参数和位置参数。以左副翼漂移故障寿命样本为例,通过极大似然估计,得到$m=1.0258$、$\eta=4.2579\times10^4$,且结果通过KS检验,所以系统不发生左副翼漂移故障的概率函数为

$$P_{trailing}(t)=e^{-\left(\frac{t}{4.2579\times10^4}\right)^{1.0258}}\tag{5}$$

对建立 AADL 飞控系统模型以左副翼漂移故障为顶事件执行故障树分析过程，得到左副翼漂移故障对应故障树如图 12 所示，各事件含义及底事件失效率如表 6 所示。

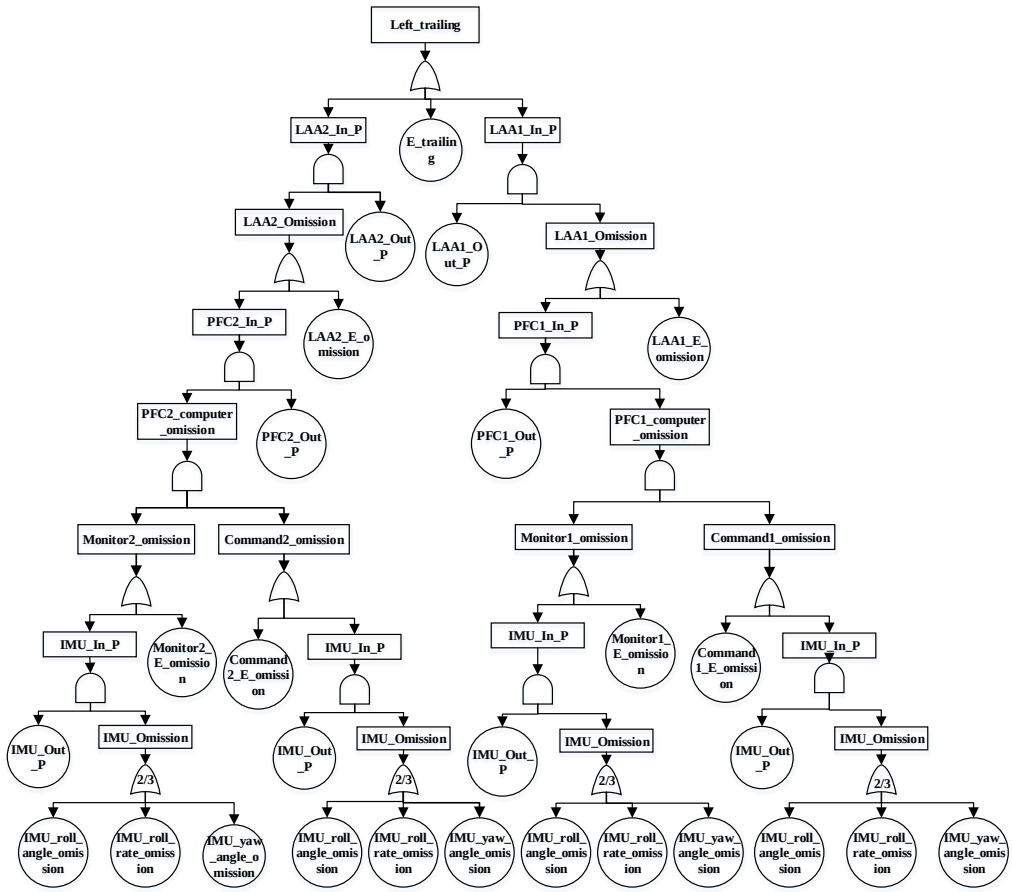



图 12 左副翼漂移故障树

Fig. 12 The fault tree of the trailing state of left aileron

表 6 故障树各事件含义

Table 6 The meaning of each event in the fault tree

| 事件名 | 含义 | 失效率(1/h) | 事件名 | 含义 | 失效率(1/h) |
|---|---|---|---|---|---|
| Left_trailing | 左副翼漂移故障 | N/A | LAA1/2_In_P | LAA 传入传播无响应错误 | N/A |
| E_trailing | 左副翼漂移错误事件 | 1e-5 | LAA1/2_Omission | LAA 无响应故障 | N/A |
| LAA1/2_Out_P | LAA 传出传播无响应错误 | 3e-4 | LAA1/2_E_Omission | LAA 无响应错误事件 | 1e-6 |
| PFC1/2_In_P | 飞控计算机传播无响应错误 | N/A | PFC1/2_computer_Omission | 飞控计算机无响应故障 | N/A |
| PFC1/2_Out_P | 飞控计算机传出传播错误 | 2e-4 | Monitor1/2_Omission | 监控通道无响应故障 | N/A |
| Command1/2_Omission | 指令通道无响应故障 | N/A | Monitor1/2_E_Omission | 监控通道无响应错误事件 | 2e-7 |
| Command1/2_E_Omission | 指令通道无响应错误事件 | 2e-7 | IMU_Out_P | IMU 传播无响应错误 | 1e-4 |
| IMU_Omission | IMU 无响应故障 | N/A | IMU_roll_angle_Omission | IMU 滚转角计算单元无响应故障 | 4e-7 |
| IMU_roll_rate_Omission | IMU 滚转率计算单元无响应故障 | 4e-7 | IMU_yaw_angle_Omission | IMU 偏航角计算单元无响应故障 | 4e-7 |

由于蒙特卡洛仿真方法通过生成随机数求解模型，求得概率会具有一定误差。在民用飞机系统较为关注平均航段时间内的故障概率如图 13 所示，取平均航段时间为 20 h，故障树分析方法中顶

事件在平均航段时间内的最大发生概率为 $1.9922\times10^{-4}$，不发生左副翼漂移故障的最小概率 $p_{ft}=0.99980078$，本文方法求得不发生左副翼漂移故障的最小概率 $p=0.99961462$，两种方法求得不发生左副翼漂移故障的最小概率均大于0.999 6，根据相对误差计算方法：$\left|p_{ft}-p\right|/\max\left(p_{ft},p\right)$，两种方法在平均航段时间内的最大相对误差为0.018%，满足工程需求。

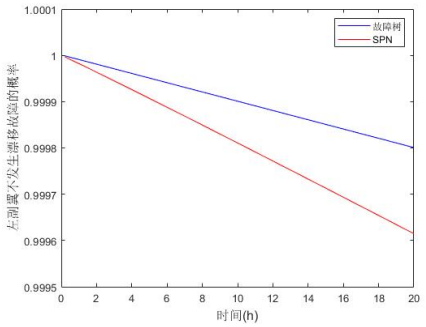



图13　平均航段时间内不发生左副翼漂移故障概率曲线

Fig. 13　The probability curve of left aileron is not in trailing state during the average segment time

## 4　结　论

1）采用AADL建立系统名义模型与拓展模型，在此基础上生成用于可靠性分析的SPN，与传统可靠性分析方法相比，能够保证可靠性模型与设计模型的一致性，并且避免了对分析人员经验和技能的依赖。

2）在AADL模型的基础上自动生成用于可靠性分析的SPN，当系统需要更改时，只需修改相应的AADL模型，不需要重复SPN构建过程，避免了传统安全性分析中构建故障树、马尔可夫模型等繁琐的建模过程。

3）使用蒙特卡洛模拟方法计算系统可靠度，可以避免建立马尔可夫模型的繁琐工作以及状态爆炸问题，提高系统可靠性计算效率。

## 参 考 文 献

[1] International S-18 CommitteeSAE. Guidelines for development of civil aircraft and systems: ARP4754A[S]. Warrendale: Society of Automotive Engineers, 2010.

[2] International S-18 CommitteeSAE. Guidelines and methods for conducting the safety assessment process on civil airborne system and equipment: ARP4761[S]. Warrendale: Society of Automotive Engineers, 1996.

[3] JOSHI A, HEIMDAHL M. Model-based safety analysis: NASA/CR-2006-213953[R]. Washington: NASA, 2006.

[4] 陈磊，焦健，赵廷弟. 基于模型的复杂系统安全分析综述[J]. 系统工程与电子技术，2017，39(6)：1287-1291.
CHEN Lei, JIAO Jian, ZHAO Tingdi. Review for model-based safety analysis of complex safety-critical system[J]. Systems Engineering and Electronics, 2017, 39(6): 1287-1291.(in Chinese)

[5] PECIAK M, SKARKA W. Assessment of the potential of electric propulsion for general aviation using model-based system engineering (MBSE) methodology[J]. Aerospace, 2022, 9(2): 74.

[6] 祁健，胡军，谷青范，等. 一种AltaRica 3.0模型中类的平展化方法[J]. 计算机科学，2021，48(5)：51-59.
QI Jian, HU Jun, GU Qingfan, et al. Class flattening method for AltaRica 3.0 model[J]. Computer Science, 2021, 48(5): 51-59.(in Chinese)

[7] 展万里，胡军，谷青范，等. 基于模型的故障树自动生成方法[J]. 计算机科学，2021，48(12)：159-169.
ZHAN Wanli, HU Jun, GU Qingfan, et al. Model-based fault tree automatic generation method[J]. Computer Science, 2021, 48(12): 159-169.(in Chinese)

[8] SANNES P S, APVRILLE L, VINGERHOEDS R. Checking SysML models against safety and security properties[J]. Journal of Aerospace Information Systems, 2021, 18(12): 906-918.

[9] FEILER P H, LEWIS B A, VESTAL S. The SAE architecture analysis & design language (AADL) a standard for engineering performance critical systems[C]// IEEE International Symposium on Computer Aided Control System Design. NY: IEEE, 2006: 302-307.

[10] SAE. Architecture analysis and design language (AADL): AS5506[S]. Warrendale: Society of Automotive Engineers, 2017.

[11] DELANGE J, FEILER P. Architecture fault modeling with the AADL error-model annex[C]// 2014 40th Euromicro Conference Series on Software Engineering and Advanced Application (SEAA 2014). NY: IEEE, 2014: 361-368.

[12] LU Zhong, ZHUANG Lu, DONG Li, et al. Model-based safety analysis for the fly-by-wire system by using Monte Carlo simulation[J]. Processes, 2020, 8(1): 90-101.

[13] 李梦蝶，赵光，罗灵鲲，等. 基于改进CNN-LSTM的飞控系统剩余寿命预测[J]. 计算机工程与应用，2022，58(16)：274-283.
LI Mengdie, ZHAO Guang, LUO Lingkun, et al. Remaining useful life prediction of flight control system based on improved CNN-LSTM[J]. Computer Engineering and Applications, 2022, 58(16): 274-283.(in Chinese)

[14] SONG Jia, ZENG Jia, LU Wentao, et al. A mission reliability analysis method of flight control system based on AltaRica language[C]// International Conference on Guid-

ance, Navigation and Control. Berlin, Heidelberg: Springer, 2022: 3405-3414.

[15] 刘畅, 蒋永平, 马春燕, 等. 基于NuSMV的AADL模型形式化验证技术[J]. 航空学报, 2022, 43(3): 451-466.
LIU Chang, JIANG Yongping, MA Chunyan, et al. Formal verification technology for AADL models based on NuSMV [J]. Acta Aeronautica et Astronautica Sinica, 2022, 43(3): 451-466. (in Chinese)

[16] 王瀚博, 周兴社, 董云卫, 等. 结构分析和设计语言AADL研究[J]. 计算机工程与应用, 2009, 45(16): 1-4.
WANG Hanbo, ZHOU Xingshe, DONG Yunwei, et al. Research on architecture analysis and design language [J]. Computer Engineering and Applications, 2009, 45(16): 1-4. (in Chinese)

[17] YUAN Cangzhou, WU Kangzhao, CHEN Guotao, et al. An automatic transformation method from AADL reliability model to CTMC[C]// 2021 IEEE International Conference on Information Communication and Software Engineering (ICICSE). NY: IEEE, 2021: 322-326.

[18] 李静, 沈宁敏, 白海洋, 等. 基于时间自动机的嵌入式系统AADL模型可调度性验证[J]. 东南大学学报(自然科学版), 2015, 45(6): 1032-1037.
LI Jing, SHEN Ningmin, BAI Haiyang, et al. Schedulability verification of embedded system AADL model based on timed automata[J]. Journal of Southeast University (Natural Science Edition), 2015, 45(6): 1032-1037. (in Chinese)

[19] JIANG Zeyong, ZHAO Tingdi, WANG Shihai, et al. New model-based analysis method with multiple constraints for integrated modular avionics dynamic reconfiguration Process [J]. Processes, 2020, 5: 10-18.

[20] 董云卫, 王广仁, 张凡, 等. AADL模型可靠性分析评估工具[J]. 软件学报, 2011, 22(6): 1252-1266.
DONG Yunwei, WANG Guangren, ZHANG Fan, et al. Reliability analysis and assessment tool for AADL model [J]. Journal of Software, 2011, 22(6): 1252-1266. (in Chinese)

[21] 莊露, 陆中, 张子文. 基于随机Petri网的机载系统动态可靠性建模[J]. 西北工业大学学报, 2020, 38(4): 846-854.
ZHUANG Lu, LU Zhong, ZHANG Ziwen. Dynamic reliability model for airborne systems based on stochastic Petri net[J]. Journal of Northwestern Polytechnical University, 2020, 38(4): 846-854. (in Chinese)

**作者简介:**

**罗文斌**(1998—),男,硕士研究生。主要研究方向:系统可靠性分析。

**陆　中**(1980—),男,博士,教授。主要研究方向:适航符合性验证,系统安全性评估。

**程大炜**(1999—),男,硕士研究生。主要研究方向:系统可靠性分析。

**缪炜润**((2000—),男,硕士研究生。主要研究方向:系统可靠性分析。

**(编辑:马文静)**