计算机科学
COMPUTER SCIENCE

基于STPA的飞行导引系统模式转换的安全性分析研究

左辰翠, 黄志球, 胡军, 谢健, 徐恒, 石帆

引用本文

左辰翠, 黄志球, 胡军, 谢健, 徐恒, 石帆. 基于STPA的飞行导引系统模式转换的安全性分析研究[J]. 计算机科学, 2026, 53(1): 341-352.

ZUO Chencui, HUANG Zhiqiu, HU Jun, XIE Jian, XU Heng, SHI Fan. Research on Safety Analysis of Mode Transition of Flight Guidance System Based on STPA [J]. Computer Science, 2026, 53(1): 341-352.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000156

基于 STPA 的飞行导引系统模式转换的安全性分析研究

左辰翠　黄志球　胡　军　谢　健　徐　恒　石　帆
南京航空航天大学计算机科学与技术学院　南京 210016
(chencuizuo@163.com)

摘　要　在民机自动飞行过程中,飞行导引系统的模式转换是影响安全的重要因素,应对其进行充分的安全性分析。传统安全分析方法主要关注各个组件的失效因素,忽略了由组件间非线性交互产生的安全问题。为此,采用系统理论过程分析(System Theory Process Analysis,STPA)方法,对飞行导引系统模式转换进行系统且完整的分析。同时,鉴于 STPA 方法中存在需人工分析的部分,引入了基于时间自动机理论的形式化模型检查工具 UPPAAL 对系统进行建模与验证,以确保控制结构图的正确性,并识别真正不安全控制行为(Unsafe Control Action,UCA),从而避免资源的浪费。最后,提出规范化的致因因素分析框架对通过验证的 UCA 进行逐一分析。实例证明,所提方法对航空类复杂系统安全性分析具有较好的效果。

关键词:飞行导引系统;模式转换;系统理论过程分析;模型检查;UPPAAL
中图分类号　V249

Research on Safety Analysis of Mode Transition of Flight Guidance System Based on STPA

ZUO Chencui,HUANG Zhiqiu,HU Jun,XIE Jian,XU Heng and SHI Fan
School of Computer Science and Technology,Nanjing University of Aeronautics and Astronautics,Nanjing 210016,China

Abstract　In the process of automatic flight of civil aircraft,the mode transition of the flight guidance system is an important factor affecting safety and should be subject to a comprehensive safety analysis. Traditional safety analysis methods mainly focus on the failure factors of individual components,ignoring the safety issues arising from the nonlinear interactions between components. For this reason,this paper adopts the System Theory Process Analysis(STPA) method to conduct a systematic and complete analysis of the mode transition of the flight guidance system. Meanwhile,considering that there are parts in the STPA method that require manual analysis,the formal model checking tool UPPAAL based on the theory of timed automata is introduced to model and verify the system,so as to ensure the correctness of the control structure diagram and identify the truly Unsafe Control Actions(UCA),thus avoiding the waste of resources. Finally,a standardized causal factor analysis framework is proposed to analyze the verified UCAs one by one. The example proves that the proposed method is effective for the safety analysis of aviation complex systems.

Keywords　Flight guidance system,Mode transition,System theoretic process analysis,Model checking,UPPAAL

1　引言

随着航空技术的迅速发展,自动飞行系统已经成为现代航空器的核心组成部分[1]。其中,飞行导引系统(Flight Guidance System,FGS)在自动飞行中起着重要作用,主要用于协助飞行员进行飞行的基本控制和战术指导。该系统会指引飞机按照预设的飞行线路或者根据环境状况选择合适的飞行控制律实现自动飞行。飞行导引系统的模式是连接飞行任务与具体飞行控制律的桥梁。然而,随着系统复杂性和自动化水平的提高,飞行导引系统模式转换过程中的安全性问题日益突出。2009 年,法航 447 号航班在高度和速度不当的情况下进行了不正确的飞行模式转换,导致飞机进入失速状态,最终坠毁于大西洋[2];1991 年,美联航 706 号航班在飞行模式转换过程中,飞行员对自动驾驶系统的误操作,导致飞机在起飞阶段偏离航线,最终导致空难发生[3]。这些事故的发生令人扼腕,因此迫切需要针对飞行导引系统模式转换进行深入的安全性分析。

当前,关于飞行导引系统的安全性分析的研究已经展开。国内外学者基于 ARP4754[4] 和 ARP4761[5] 中的传统安全分析方法对 FGS 展开了一系列安全性分析。传统的安全性分析方法有基于故障树模型(Fault Tree Analysis,FTA)[6]、失效模式与影响分析模型(Failure Mode and Effects Analysis,FMEA)[7] 等方法。Miller 等[8] 在美国宇航局兰利研究中心等组织的支持下,基于 RSML^{-e} 语言为 FGS 模式逻辑编写了需求规范;随后 Tribble 等[9] 基于该需求规范将 FTA 和 FMEA 分析得到的基本事件和失效模型的列表作为 FGS 的

到稿日期:2024-10-28　返修日期:2025-03-10
基金项目:国家自然科学基金联合基金(U224120044)
This work was supported by the Joint Funds of the National Natural Science Foundation of China(U224120044).
通信作者:黄志球(zqhuang@nuaa.edu.cn)

安全性质，并使用模型检查方法对安全性质进行了形式化验证；Joshi 等[10]通过 PVS 定理证明器[11]和 NuSMV 模型检查器[12]等形式化方法[13]研究了 FGS 内部模态混淆问题。然而，传统 FTA 和 FMEA 安全性分析方法仅关注组件失效以及失效之间的因果关系，认为事故仅由线性事件链导致，忽视了系统/子系统(组件)/环境之间不安全交互产生的安全风险问题[14]。随着航空器自动化程度的提高，飞行导引系统逐步实现自动化，组件间的交互愈加复杂，传统安全性分析方法难以进行全面的分析，亟需更有效的方法来填补这一研究空白。

为了对系统/子系统/环境之间的交互行为进行安全性分析，美国麻省理工学院的 Leveson 等[15-16]基于系统理论提出了系统理论事故模型(System Theoretic Accident Model and Processes，STAMP)，并基于此提出了系统理论过程分析(STPA)方法。系统理论为理解复杂系统的运行机制提供了全面的视角。STPA 基于系统理论，通过关注系统组件间的控制与反馈关系来识别危害。它不局限于传统的故障模式分析，而是深入探究系统在正常运行过程中可能因组件交互引发的危险状况。国内外基于 STPA 方法在不同的安全性分析领域已经开展了大量的研究工作[17-21]。这些研究充分证明了基于成熟系统理论的 STPA 方法在应对复杂系统安全性分析时的有效性与优越性。

尽管 STPA 在风险识别和系统安全性分析方面具有显著优势，但仍然存在一些局限性。Tsuji 等[22]针对 STPA 方法分析出的损失场景数量庞大问题，提出了一种利用统计模型检查技术对损失场景进行优先级排序的方法。De Souza 等[23]针对 STPA 方法缺乏形式化的问题，在支持系统工程建模语言 SysML 的免费软件 TTool 中进行控制结构模型搭建，然后使用模型检验器 UPPAAL 进行形式化验证。Zhong 等[24]针对 STPA 分析效率低下的问题，提出了一种 STPA-SN 框架，在 SysML 中建立模型，用 MARTE 描述时序，最后使用模型检查器将 SysML 模型自动转换为 NuSMV 模型并输出损失场景，提高了分析效率。本文主要探讨两点局限性：首先，传统 STPA 方法难以保证依赖领域专家手动搭建的控制结构图符合系统预期的功能设定，即模型正确性；其次，由领域专家分析或根据枚举法[25]自动生成的 UCA 的有效性难以保证，后续的损失场景分析将造成人力资源浪费。针对这些局限性，本文提出了一种将 STPA 与 UPPAAL 相结合的分析框架，并给出了从 STPA 模型向用于模型检验的形式化模型的系统转换过程。通过两次模型检验，确保模型的正确性与 UCA 的有效性，提高分析的质量与效率。为了展示所提出的框架，本文选用了 NASA 飞行导引系统模式转换案例进行研究，证实了该方法框架的可行性。

本文的主要贡献如下：

1)使用 STPA 方法对飞行控制系统的模式转换问题进行系统的安全性分析；

2)针对目前 STPA 在模型正确性与 UCA 的有效性难以保证的局限性，引入基于时间自动机理论的形式化模型验证工具 UPPAAL[26]对 STPA 进行扩展；

3)使用规范化的致因因素分析框架对通过验证的 UCA 进行系统的致因因素和损失场景分析，实例证明，所提方法对航空类复杂系统的安全性分析具有较好的效果。

2 飞行导引系统

2.1 基本信息

飞行导引系统是飞行控制系统(Flight Control System，FCS)的重要组成部分，它负责从 AHRS，ADS 和 Nav Radio 这些传感器中收集飞机当前的飞行状态以及各种环境数据，将测量状态与来自飞行员或飞行管理系统(Flight Management System，FMS)设定的期望状态进行比较，并向自动驾驶仪(Autopilot，AP)和飞行指引仪(Flight Director，FD)提供俯仰/滚转制导指令，最小化测量状态与期望状态之间的差异。图 1 给出了强调 FGS 作用的 FCS 简化概述图，显示了 FCS 中与 FGS 工作有着紧密联系的一些系统接口。

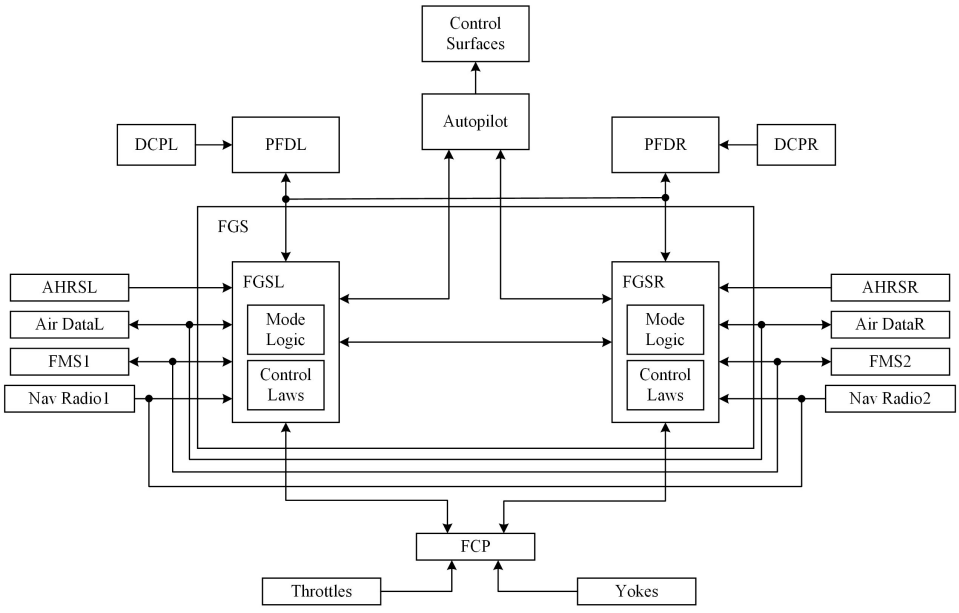


图 1 飞行控制系统概览图

Fig. 1 Overview of flight control system

飞行控制面板(Flight Control Panel，FCP)和主飞行显示器(Primary Flight Display，PFD)是飞行员与飞行导引系统模式之间的关键人机接口组件。飞行员可以通过 FCP 上的按钮选择所需的飞行模式，对 AP 和 FD 进行开闭操作。而 PFD 需要及时、准确地向飞行员展示各种与飞行模式相关的信息，包括飞行指引命令、俯仰和滚转制导指令、飞行模式状态，以及自动驾驶仪的工作端状态等。

FGS 包含模式逻辑(Mode Logic)[27]和飞行控制律(Flight Control Laws)[28]两个部分。飞行控制律是将飞机当前状态与期望状态进行比较并生成缩减两者差异的连续函数，而模式逻辑是在某些模式处于激活态时，用于选择正确飞行控制律的离散算法。

FD 根据选定的工作模式自动计算操纵指令，并指导飞行员操作。它将实际飞行路径与目标飞行路径进行比较，计算出到达目标航向所需要的操纵量，并以目视的形式在 PFD 上显示操纵量。与 AP 不同的是，FD 不直接控制飞机，而是对飞行员进行“指挥”。

2.2 模式逻辑基础

飞行模式可以定义为一组互斥的系统行为。飞机在不同的阶段使用不同的飞行模式实现对飞机航向、高度和速度的控制。针对飞行导引系统的工作模式标准，AC25.1329 咨询通告[29]对其进行了规范说明，该通告将飞机的飞行模式规范分为 4 种：横向模式(Lateral Modes)、垂直模式(Vertical Modes)、多轴模式(Multi-Axis Modes)和自动油门模式(Auto-thrust Modes)。具体划分如图 2 所示。其中多轴模式是指由多种飞行模式共同作用完成飞行任务的模式。

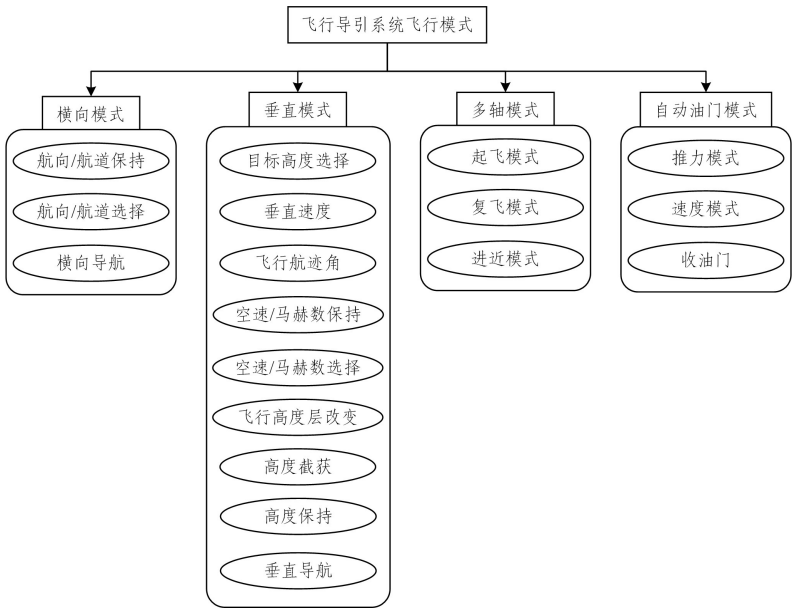


图 2 AC25.1329 飞行导引系统模式划分

Fig. 2 Classification of flight guidance system mode in AC25.1329

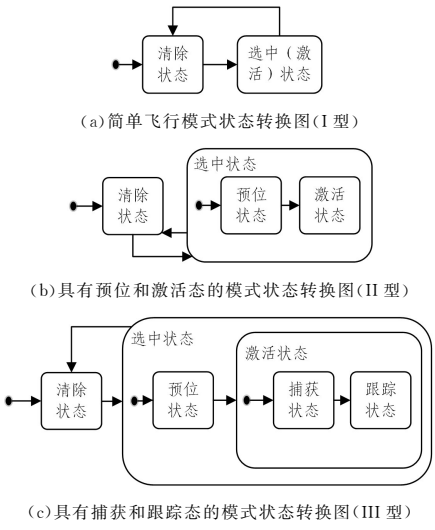
飞行模式逻辑由飞行模式和模式状态转换条件两个部分组成。最简单的模式只包含两个状态，即清除和选中，如图 3(a)所示。这种模式一旦被选中将立即被激活，模式逻辑模块将调用相关的飞行控制律，生成的制导指令将被提供给 AP 和 FD。当模式被清除时，该模式的相关飞行控制律处于非操作状态，即不会产生任何输出。

一些模式可以在满足某些条件时，如获得导航源或者目标参考(如期望高度)，从预位变成激活。这种模式如图 3(b)所示，预位和激活是选中状态的两个子状态。在预位状态下，该模式相关的飞行控制律并不会产生相关的制导指令，但它可能会接受输入以及积累状态信息，并帮助确定是否满足激活模式的条件。一旦条件满足，模式将立即从预位态转为激活态，其相关飞行控制律立即向 AP 和 FD 发送制导指令。需要注意的是，退出激活态的唯一方式就是取消选择该模式，直接回到清除态。

此外，一些模式还区分了目标参考或导航员的捕获和跟踪。这种模式如图 3(c)所示，一旦处于激活态，该模式的飞行控制律首先通过操纵飞机使其与参考点或导航源进行对齐来捕获目标。一旦正确对齐，模式将转入跟踪态，在这种状态下，它将飞机保持在目标上。捕获态和跟踪态都属于激活态，飞行控制律都会向 AP 和 FD 提供制导指令。同样，退出预位、捕获和跟踪态的唯一方法就是取消选择该模式，直接回到清除态。

飞行导引系统由模式逻辑和飞行控制律组成，FGS 接受飞行员或 FMS 的指令，在自动飞行过程中选择不同的飞行模式进行工作。模式逻辑又由四大类，至少 18 种飞行模式以及各模式中不同状态之间的转换规则构成。不同模式的状态数量不同，它们的激活条件也千差万别。

此外，飞行过程中每一时刻都至少需要两个类别的飞行模式合作进行飞行指导，各模式之间又存在着排斥与兼容关系。例如，为了给 AP 和 FD 提供明确的指导，任何时刻只能有一种横向模式和一种垂直模式处于激活状态。同样地，如果 AP 或 FD 被打开，那么至少要有一种横向模式和一种垂直模式处于激活状态。FGS 中单模式逻辑模块就已经非常复杂，各模式交互过程中难免会出现各种安全危险，因此安全性分析迫在眉睫。



(a)简单飞行模式状态转换图(I型)

(b)具有预位和激活态的模式状态转换图(II型)

(c)具有捕获和跟踪态的模式状态转换图(III型)

图3 飞行模式状态转换图

Fig. 3 Flightmode state transition diagram

3 STPA的形式化扩展

STPA通过构建系统的安全控制结构，描述系统组件之间的控制与反馈回路，进而分析这些回路之间产生的行为或数据交互所导致的潜在危害，并提出相应的安全约束。此外，针对识别出的危害，STPA还会通过一套完整的分析流程识别其根本原因和损失场景。STPA的安全性分析主要有4个步骤[30]。

1)定义分析的目的。这一步骤包括识别损失、系统级危险以及系统级安全约束。

2)构建分层控制结构。控制结构是由控制回路组成的系统功能模型，一个控制结构至少要有控制器、控制行为、反馈、受控过程和来自其余组件的输入输出。较为复杂的系统还包含执行器和传感器。

3)识别不安全控制行为。不安全控制行为(UCA)是指，在特定背景和最坏情况下会导致危险的控制行为。

4)确定损失场景。在这一步骤中，需要对每一个确定的UCA用一套完整的分析流程进行损失场景识别，并给出消除或者降低这些损失的建议。

为更好地分析以飞行导引系统模式转换为代表的复杂系统的安全性，本文选用基于系统理论的STPA方法作为分析主体。针对STPA方法固有的局限性，引入基于时间自动机理论的UPPAAL对STPA进行扩展，扩展方法框架如图4所示。图中上半部分展示了STPA方法分析的流程，下方则是UPPAAL模型的检验部分。当使用STPA对系统结构进行完整的分析并构建出控制结构模型后，按照对应规则将其转换成UPPAAL模型，以便对控制结构进行正确性验证，确保模型可达且满足系统的功能性需求。验证通过的结构模型可用于识别潜在的不安全控制行为(Potential Unsafe Control Action，PUCA)，这些PUCA将被转换为CTL公式进行验证筛除，通过验证的成为UCA，表示有概率发生此类不安全控制行为。这种筛除行为旨在为最后的损失场景与致因因素识别节省人力物力资源，提高分析效率与准确率。

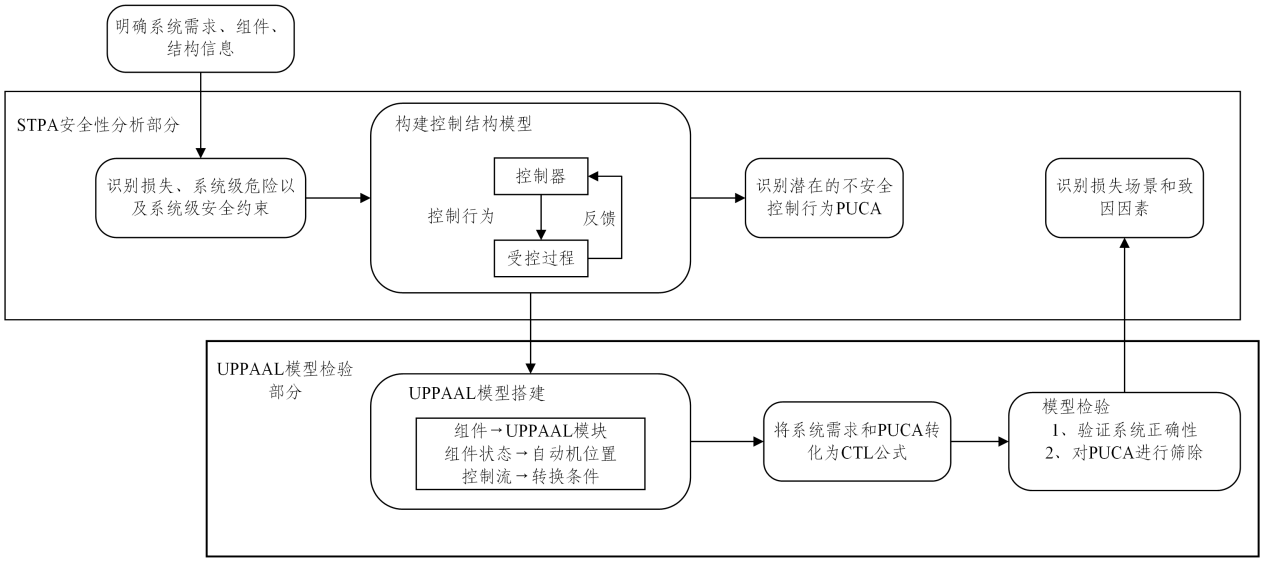


图4 形式化扩展后的STPA安全分析框架

Fig. 4 Safety analysis framework of formal extended STPA

4 飞行导引系统模式转换安全性分析实施

AC25.1329中的分类涉及18种飞行模式，不同的模式在不同的飞行阶段以特定的转换规则切换状态用于飞行工作。此外，部分模式的状态转换条件高达十多种，分析规模庞大。因此，本文选用了NASA文档中的一架典型喷气式飞机案例[8]进行方法展示。该案例重点描述了与绕轴运动相关的横向与垂直模式，如表1所列。飞行导引系统模式转换分析的复杂性在于模式状态的转换规则、激活条件以及模式间的互斥兼容关系，这些在案例中均被纳入考虑，因此该案例能够充分代表实际飞行导引系统的复杂性。实际飞行导引系统分为两个侧面，左右系统的切换也会影响模式的转换，但本文重点探讨模式之间的相互影响，因此与该因素相关的转换规则在后期完善阶段进行分析。

表 1 典型 FGS 的飞行模式

Table 1 Flight modes of typical FGS

横向模式	垂直模式
滚转保持模式 ROLL(I 型)	滚转保持模式 PITCH(I 型)
航向选择模式 HDG(I 型)	垂直速度模式 VS(I 型)
导航模式 NAV(II 型)	飞行高度改变模式 FLC(I 型)
	高度保持模式 ALT(I 型)
横向进近模式 LAPPR(II 型)	垂直进近模式 VAPPR(II 型)
横向复飞模式 LGA(I 型)	垂直复飞模式 VGA(I 型)

4.1 定义分析的目的

在第一步中，基于 STPA 的初步分析确定系统的目标、可能的危险和事故。本文的目标系统是支持自动飞行的现代客机中负责进行模式转换的飞行导引系统。飞行导引系统接收到模式转换指令后，会调动不同的飞行模式生成相应的俯仰和滚转制导指令。在这些飞行模式的作用下，飞机可以完成一次完整的飞行活动。在进行飞行模式转换的过程中，可能会导致 3 种不同的事故：人员伤亡(A1)、飞机受损(A2)、飞行任务失败(A3)。结合对系统边界的认知与飞行导引系统飞行模式转换相关知识的理解，可以识别出以下相关系统级危险。

H1：飞机在飞行过程中违反最低间隔标准。

H2：飞机爬升或下降到预选高度外。

H3：飞机的飞行速度超过最高限速。

H4：飞机偏离目的地或迷航。

H5：飞机脱离特定滑行道、跑道或地面机坪。

H6：飞行模式混乱。

4.2 构建分层控制结构

安全控制结构是由一个或者多个控制反馈回路组成的，系统的不同组件通过相关的控制命令和反馈信息进行交互。在 FGS 系统中，当飞行模式由飞行员实时设定时，飞行员通过飞行控制面板上的按钮对飞行导引系统进行控制，并通过面板上的显示灯以及主飞行显示器上显示的信息获取系统的反馈信息；当飞行模式由飞行管理系统中事先设定的飞行计划控制时，FMS 会在满足特定的条件时，向 FGS 发送模式打开/关闭信号。此外，FGS 还承担着控制自动驾驶仪和飞行指引仪开关的责任，飞行员通过 FCP 向 FGS 发送 AP 和 FD 打开/关闭指令，FGS 内部相关模块对 AP 和 FD 进行控制。

控制器 FGS 接收到飞行员或 FMS 的飞行模式控制信号后，内部的模式逻辑和飞行控制律即刻开始工作。模式逻辑向飞行控制律发送各飞行模式的状态，飞行控制律将生成被激活模式相关的制导指令，并将制导指令传输给执行器 AP 和 FD，用于后续的飞行指引。部分飞行模式需要飞机处在恰当的状态才能被激活，因此这些飞机的状态信息将通过一些传感器反馈给 FGS 系统。部分飞机状态信息需要实时反馈给飞行员进行数据监测，以便在突发情况时可以及时进行人为干预。FGS 系统进行飞行模式转换的安全性分析控制结构图如图 5 所示。

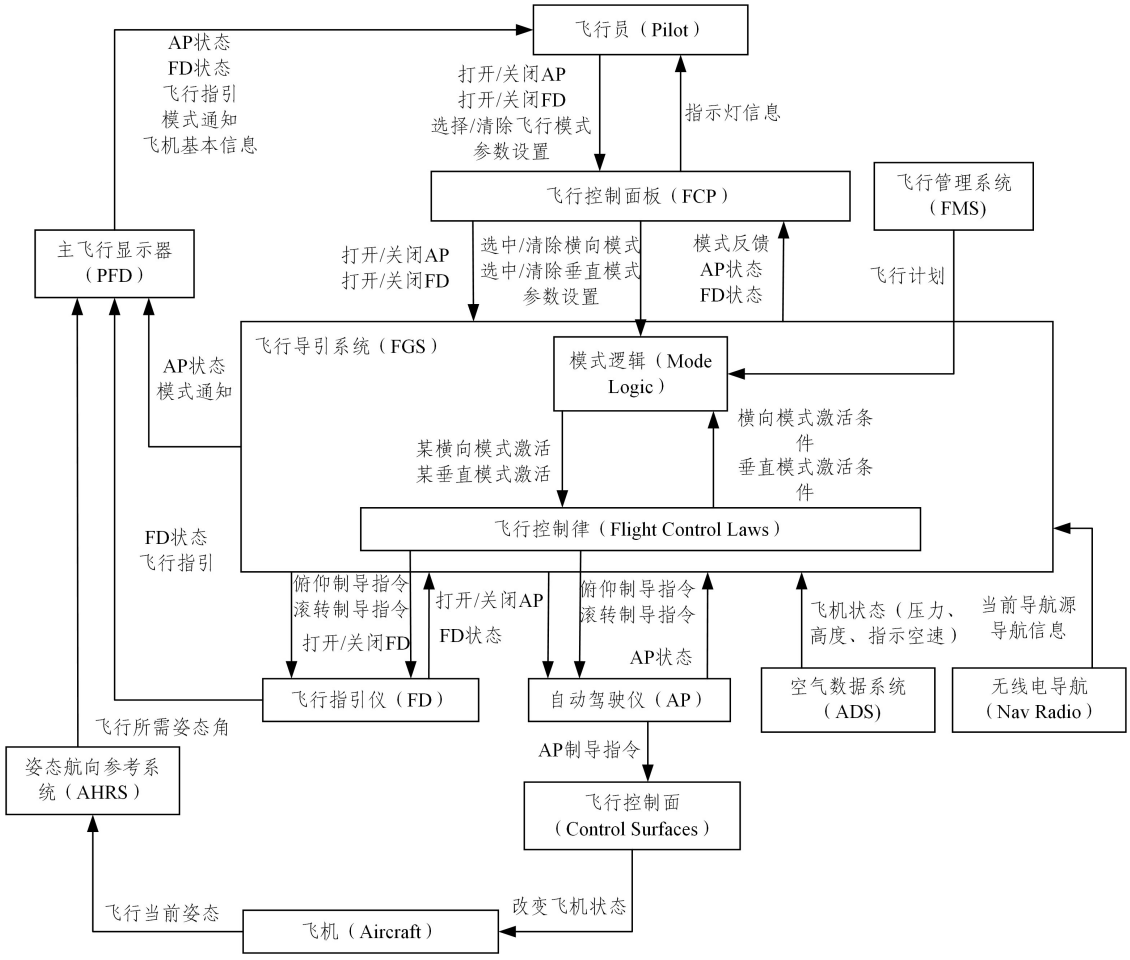


图 5 飞行导引系统模式转换控制结构图

Fig. 5 Control structure diagram of flight guidance system mode transition

4.3 识别不安全控制行为

在 STPA 方法中，危险的产生被认为是由实施不安全控制行为(UCA)造成的。Leveson 将 UCA 定义为在特定背景和最坏情况下会导致危险的控制动作，一般分为 4 类[30]：

1)没有提供控制动作导致的危险；

2)提供有误的控制动作导致的危险；

3)提供了相应的控制动作，但提供的时机不当(过早、过晚或顺序错误)导致的危险；

4)提供的控制行为持续时间过长或停止过早导致的危险(此类针对连续型控制行为)。

本文认为 STPA 初步分析出的不安全控制行为未经验证，无法证实其一定会发生，因而先将本节分析出的不安全控制行为定义为潜在的不安全控制行为(PUCA)，当这些 PUCA 经过验证会在某种情况下发生时才是 UCA。初步分析出的部分 PUCA 如表 2 所列。

表 2 部分潜在的不安全控制行为

Table 2 Partial potential unsafe control action

控制行为	未提供控制行为	提供有误控制行为	控制行为提供节点有误	控制行为时长有误
俯仰制导指令	PUCA-1:FGS 在飞行员/飞行计划选择了 VS 模式以设定好的垂直速度进行爬升或下降时未提供 VS 模式相关的俯仰制导命令(H2) PUCA-2:FGS 在飞行员/飞行计划选择了 FLC 模式进行爬升或下降到指定高度时未提供 FLC 模式相关的俯仰制导命令(H2) PUCA-3:FGS 在飞机处于超速状态时，未选择 FLC 模式提供相关俯仰制导指令进行速度调整(H1,H3) PUCA-4:FGS 在飞行员/飞行计划选择了 ALT 模式保持参考高度进行飞行时未提供 ALT 模式相关的俯仰制导命令(H2) PUCA-5:FGS 在飞行员/飞行计划选择了 APPR 模式进行进近着陆且检测到跟踪条件已被满足时未提供 VAPPR 模式相关俯仰制导指令(H5) PUCA-6:FGS 在飞行员/飞行计划选中了 GA 模式进行复飞时未提供 VGA 模式相关俯仰制导指令(H4)	PUCA-7:FGS 在飞机正处于超速状态时提供了 VS/VGS 模式相关制导指令(H1,H4) PUCA-8:FGS 在飞机选择了其他垂直模式进行工作时仍然提供了 PITCH/VS/FLC/ALT/VAPPR/VGA 模式相关俯仰制导指令(H6)	PUCA-9:FGS 在 LAPPR 模式被激活提供相关滚转制导指令之前已经开始提供 VAPPR 模式相关俯仰制导指令(H2,H5)	N/A
滚转制导指令	PUCA-10:FGS 在飞行员/飞行计划选择了 HDG 模式保持当前航向进行飞行时未提供 HDG 模式相关滚转制导命令(H4) PUCA-11:FGS 在飞行员/飞行计划选择了 NAV 模式跟踪航路导航和进行非精确性进近，且跟踪条件已被满足时未提供 NAV 模式相关滚转制导指令(H4,H5) PUCA-12:FGS 在飞行员/飞行计划选择了 APPR 模式进行进近着落且跟踪条件已被满足时未提供 LAPPR 模式相关滚转制导指令(H5) PUCA-13:FGS 在飞行员/飞行计划选择了 GA 模式进行复飞时未提供 LGA 模式相关滚转制导指令(H4)	PUCA-14:FGS 在飞机处于超速状态时仍提供了 LGA 模式相关滚转制导指令(H1,H4) PUCA-15:FGS 在飞行员/飞行计划选择了其他横向模式进行工作时仍提供了 ROLL/HDG/NAV/LAPPR/LGA 模式相关滚转制导指令(H6)	—	—

4.4 UPPAAL 建模及形式化验证

本文选用基于时间自动机理论的形式化建模工具 UPPAAL 进行系统形式化验证。当 UPPAAL 模型建立之后，采用巴科斯-瑙尔范式(Backus-Naur Form，BNF)语句验证时间自动机网络[31]，验证主要包括系统正确性验证以及不安全控制行为验证两个方面。

4.4.1 建模方法

时间自动机[26]是一个六元组(L,l_0,C,A,E,I)。其中：L 是位置集；$l_0\in L$ 是初始位置；C 是时钟集；A 是动作、协作动作和内部 τ-动作的合集；$E\subseteq L\times A\times B(C)\times 2^C\times L$ 是具有动作、卫士条件和待重置时钟集的位置之间的状态迁移集合，$B(C)$表示该位置时钟必须满足的条件；$I:L\to B(C)$为位置分配不变时钟条件。

时间自动机网络由一组时间自动机组成，这组时间自动机之间共享时钟和动作集。不同的时间自动机之间通过握手进行同步通信，可用于模拟系统内组件间的控制与反馈关系。声明一个同步通道 c，用 c! 表示发出同步信息，用 c? 表示接收同步信息。通过发送、接收同步信息，将接收到同步信息的时间状态机转移到下一个状态。

4.4.2 系统建模

根据飞行导引系统的模式转换控制结构图以及图中组件的内部结构和信息建立 UPPAAL 模型，模型中元素的命名参考 NASA 文档。控制结构图中涉及到的组件均对应一个或多个时间自动机，这些时间自动机共同构成飞行导引系统的时间自动机网络。

组件或模式的状态构成自动机的位置集。初始位置是自动机开始时所处的位置。FGS 中，对时钟的运用主要在于 FCP 中各指令的优先级表达，使用限时接收信号的方式决定响应指令。FCP 中的时钟构成时钟集。用于表示控制和反馈的同步以及组件或模式的内部行为构成了动作集。状态转换及其转换条件构成了自动机的状态迁移。本文并未使用不变量对位置的状态进行约束。

根据控制结构图中的组件确定好需要搭建的时间自动机后，需要构建时间自动机网络中自动机间的动作交互。以 Pilot 和 FCP 间的控制反馈关系为例，结构图中 Pilot 对 FCP 有一个控制操作“打开 AP”，因此声明一个同步通道“AP_Engage_Pressed”，Pilot 自动机从 idle 位置发送同步“AP_Engage_Pressed!”，由于 Pilot 未发生状态变化，因此终位置也是

idle。相应地，FCP 自动机需要从 idle 位置接收该同步“AP_Engage_Pressed?”到达“AP_Press_Button”位置，实现状态迁移。整个系统中的控制回路均按照该步骤搭建。

为了更清晰地展示建模过程，下文按照组件的功能分类逐一展示模型。

1）控制器建模

飞行导引系统包含两个控制器，即飞行员 Pilot 和飞行导引系统 FGS。Pilot 是用于表示飞行员对系统下达指令的抽象组件，属于高层级控制器。FGS 是系统的核心，管控着飞行模式和飞行控制律的选择。

飞行员通过 FCP 上的按钮控制 FGS 中模式的选择，由于人脑判断的复杂性，省略了 Pilot 的状态转换，仅用于发送同步信号，如图 6 所示。

FGS 中由于涉及的飞行模式以及对应的飞行控制律数量较多，且各模式间的转换与兼容互斥关系极为复杂，因此使用多个自动机进行状态模拟。FGS 的建模主要包括 3 个部分：专用于向各模式通知整个模式开关状态的自动机、各横向模式和垂直模式专属自动机，以及飞行控制律调用自动机。

在模型中，本文使用了模式这一抽象概念。此概念仅表示是否需要飞行模式进行工作。该组件含打开和关闭两个状态。只有模式打开时，各模式接收到打开信号才开始进行状态转换；一旦模式被关闭，所有模式立即停止工作。因此，单独为模式建立一个状态机，统筹所有飞行模式的开闭，如图 7 所示。

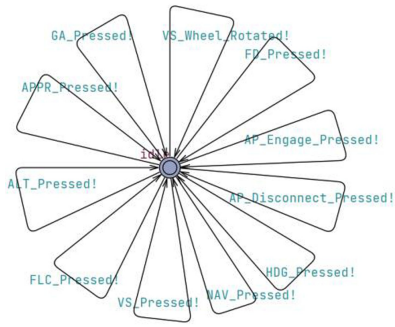


图 6 飞行员时间自动机

Fig. 6 Automaton of Pilot

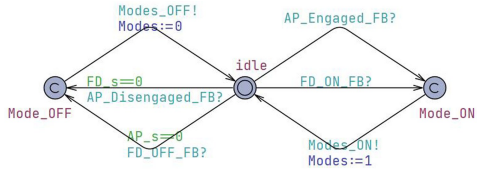


图 7 飞行模式自动机

Fig. 7 Automaton of flight modes

为了便于表现各模式之间的互斥关系，每个横向和垂直模式都将单独构建一个状态机，如图 8 和图 9 所示。模式状态以及状态转换信息从 NASA 文档中获取。

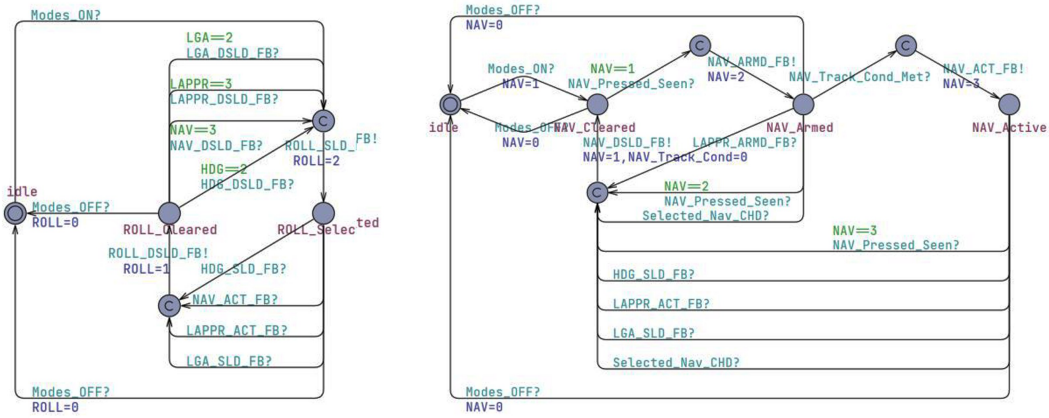


图 8 部分横向模式时间自动机

Fig. 8 Partial automatons of lateral modes

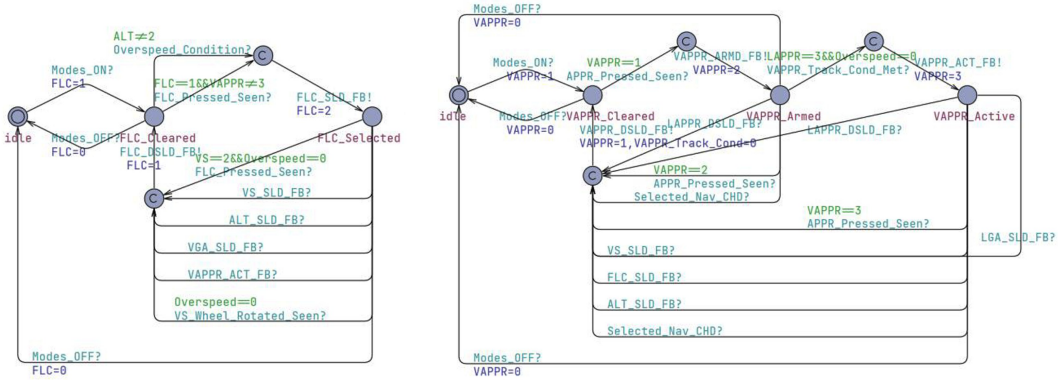


图 9 部分垂直模式时间自动机

Fig. 9 Partial automatons of vertical modes

飞行控制律自动机(见图 10)展示了根据模式逻辑模块激活的模式选择相应的飞行控制律,生成的制导指令会传送给 AP 和 FD 进行飞行指引和飞行控制。同样地,分别对横向模式和垂直模式进行建模。

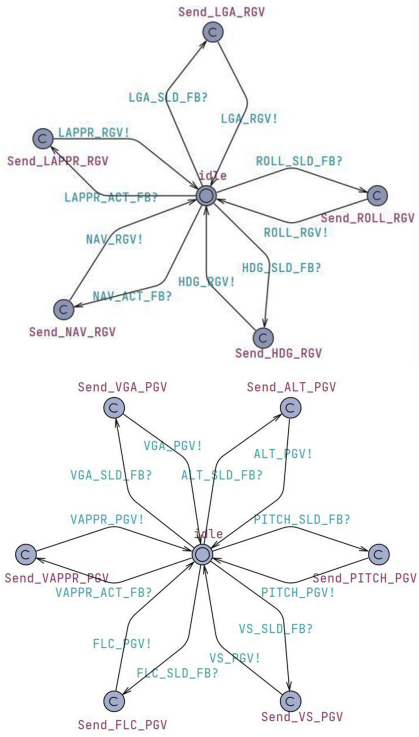


图 10 飞行控制律时间自动机

Fig. 10 Automatons of flight control laws

2)执行器建模

飞行导引系统包含 3 个主要的执行器:自动驾驶仪 AP、飞行指引仪 FD 和飞行控制面板 FCP。自动驾驶仪自动机(见图 11)和飞行指引仪自动机(见图 12)的构建原理类似,根据 NASA 文件中展示的开闭条件进行状态转换,在到达每个状态之前增加一个 committed 状态和状态反馈同步信号,目的是及时通知其余相关自动机 AP 与 FD 的状态反馈,降低信号延迟风险。

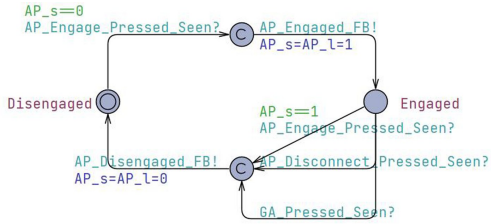


图 11 自动驾驶仪时间自动机

Fig. 11 Automaton of autopilot

嵌入式系统中常常会出现的一个现象:多个事件在同一时间发生。NASA 提出的解决方法是设定事件的优先级,当多个指令在同一时间被下达,FCP 会根据优先级序列决定高优先级的指令被响应。

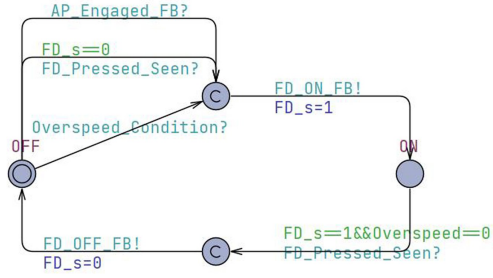


图 12 飞行指引仪时间自动机

Fig. 12 Automaton of flight director

由于 AP 开闭、横向模式选择和垂直模式选择自成一条优先级序列,这 3 条序列不存在直接的冲突关系,因此本文为 AP 开闭、FD 开闭、横向模式选择和垂直模式选择 4 个部分分别构建 FCP 时间自动机(见图 13)。

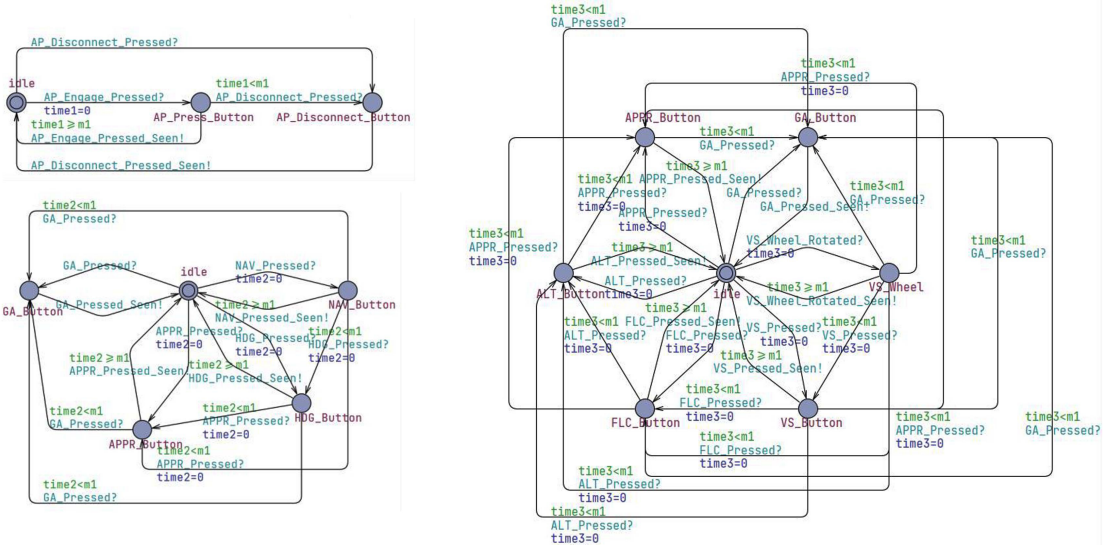


图 13 飞行控制面板时间自动机

Fig. 13 Automatons of FCP

这样做的好处是可以在保证互斥事件优先级的同时,可兼容事件的处理不受干扰。“同时”是具有抽象意义的概念,故使用时钟变量设定一个响应时间,从而表示“同时”。m1 表示一条指令被传输到 FCP 到被处理的最大时间限制,若时钟

变量小于 m1 时，有更高优先级的指令被下达，则原指令被丢弃，时钟变量重新开始为新指令计时；否则，该指令被响应，传输到 FGS 中。

3）传感器建模

系统中还有许多用于接收环境或飞机信息并将这些信息传输给控制器的传感器。传感器的建模较为简单，只需要考虑数据接收与发送即可。图 14 展示了 ADS 的时间自动机。

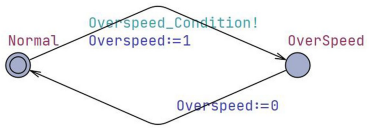


图 14 空气数据系统时间自动机

Fig. 14 Automaton of ADS

4.4.3 系统正确性验证

系统建模完成后，需要验证模型是否符合预期的功能设定。系统正确性验证包括检验模型的所有状态是否均可达，需求文档中的功能是否基本实现等。当所有的控制反馈动作都包含在模型中，自动机的状态都是可访问的，自动机的同步是一致的，并且模型没有出现死锁时，则认为建模活动成功。

UPPAAL 验证是通过验证器实施的。验证原理是利用时序逻辑公式在时间自动机网络中进行穷举搜索，查询语言采用的是分支时序逻辑 CTL[32] 的简化版本。与 CTL 类似，查询语言由路径公式（Path Formula）和状态公式（State Formula）组成。

路径公式包括：$E\langle\rangle p$ 表示存在一条路径，p 在该路径下某一状态为真；$E[]p$ 表示存在一条路径，p 在该路径下所有状态均为真；$A\langle\rangle p$ 表示对于所有路径，p 在任一路径下某一状态为真；$A[]p$ 表示对于所有路径，p 在任一路径下所有状态均为真；$p->q$ 表示当 p 成立时，q 将成立。

状态公式由 BNF 表达式表示，可以在不考虑模型行为的情况下针对某个状态进行评估。部分验证结果如表 3 所列。

表 3 系统正确性验证

Table 3 System correctness verification

验证性质	BNF 表达式	结果
系统无死锁	A[] not deadlock	True
FD 被打开，飞行模式被启动，初始横向模式为 ROLL，垂直模式为 PITCH	A<> FD. ON imply(FGS. Mode_ON and MLROLL. ROLL_Selected and MLPITCH. PITCH_Selected)	True
AP 被打开，飞行模式被启动，初始横向模式为 ROLL，垂直模式为 PITCH	A<> AP. Engaged imply(FGS. Mode_ON and MLROLL. ROLL_Selected and MLPITCH. PITCH_Selected)	True
模式打开状态下，任一时刻都至少有一个横向模式被激活	A[]!（Modes==1 and MLROLL. ROLL_Cleared and MLHDG. HDG_Cleared and（MLNAV. NAV_Cleared or MLNAV. NAV_Armed）and（MLLAPPR. LAPPR_Cleared or MLLAPPR. LAPPR_Armed）and MLLGA. LGA_Cleared）	True
模式打开状态下，非基本横向模式都不激活，那么 ROLL 模式被激活	A<>（Modes==1 and MLHDG. HDG_Cleared and（MLNAV. NAV_Cleared or MLNAV. NAV_Armed）and（MLLAPPR. LAPPR_Cleared or MLLAPPR. LAPPR_Armed）and MLLGA. LGA_Cleared）imply MLROLL. ROLL_Selected	True
模式打开状态下，任一时刻都至少有一个横向模式被激活	A[]！（Modes==1 and MLPITCH. PITCH_Cleared and MLVS. VS_Cleared and MLFLC. FLC_Cleared and MLALT. ALT_Cleared and（MLVAPPR. VAPPR_Cleared or MLVAPPR. VAPPR_Armed）and MLVGA. VGA_Cleared）	True
模式打开状态下，非基本垂直模式都不激活，那么 PITCH 模式被激活	A<>（Modes==1 and MLVS. VS_Cleared and MLFLC. FLC_Cleared and MLALT. ALT_Cleared and（MLVAPPR. VAPPR_Cleared or MLVAPPR. VAPPR_Armed）and MLVGA. VGA_Cleared）imply MLPITCH. PITCH_Selected	True
多个模式同时被选中，优先级更高的模式被响应	E<>（FCPVM. VS_Button and FCPVM. time3<m1 and FCPVM. FLC_Button and MLFLC. FLC_Selected）	True
飞行员关闭 AP 和 FD，飞行模式被关闭	E<>（AP. Disengaged && FD. OFF）implyModes==0	True

4.4.4 不安全控制行为验证

本小节将第 3 章识别出的 PUCA 转换为可用于验证的查询语句，验证结果如表 4 所列。验证通过，说明至少存在一条路径使得 PUCA 发生，那么此条 PUCA 可以被标记为真正的 UCA；验证不通过，则说明不存在任何一条路径使得 PUCA 发生，可以将此条 PUCA 进行保留，先不做分析。

表 4 不安全控制行为可达性验证

Table 4 UCA reachability verification

PUCA	BNF 表达式	结果	UCA
PUCA-1	E<>（Modes==1 and FCPVM. VS_Button and !MLVS. VS_Selected and VS<=1 and !FCLVM. Send_VS_PGV）	True	UCA-1
PUCA-2	E<>（Modes==1 and FCPVM. FLC_Button and !MLFLC. FLC_Selected and FLC<=1 and !FCLVM. Send_FLC_PGV）	True	UCA-2
PUCA-3	E<>（Modes==1 and ADS. OverSpeed and !MLFLC. FLC_Selected and FLC<=1 and !FCLVM. Send_FLC_PGV）	True	UCA-3
PUCA-4	E<>（Modes==1 and FCPVM. ALT_Button and !MLALT. ALT_Selected and FLC<=1 and !FCLVM. Send_ALT_PGV）	True	UCA-4
PUCA-5	E<>（Modes==1 and FCPVM. APPR_Button and VAPPR_Track_Cond==1 and !MLVAPPR. VAPPR_Active and VAPPR<=2 and !FCLVM. Send_VAPPR_PGV）	True	UCA-5

（续表）

PUCA	BNF 表达式	结果	UCA
PUCA-6	E()(Modes==1 and FCPVM. GA_Button and !MLVGA. VGA_Selected and VGA<=1 and !FCLVM. Send_VGA_PGV)	True	UCA-6
PUCA-7	E()(Modes==1 and ADS. OverSpeed and MLVS. VS_Selected and VS==2 and FCLVM. Send_VS_PGV)	False	—
PUCA-8	E()(Modes==1 andFCPVM. VS_Button and MLFLC. FLC_Selected and FLC==2 and FCLVM. Send_FLC_PGV)	True	UCA-7
PUCA-9	E()(Modes==1 and FCPVM. APPR_Button and FCPLM. APPR_Button and MLVAPPR. VAPPR_Active and VAPPR==3 and !MLLAPPR. LAPPR_Active and LAPPR<3)	False	—
PUCA-10	E()(Modes==1 and FCPLM. HDG_Button and !MLHDG. HDG_Selected and HDG<=1 and !FCLLM. Send_HDG_RGV)	True	UCA-8
PUCA-11	E()(Modes==1 and FCPLM. NAV_Button and NAV_Track_Cond==1 and !MLNAV. NAV_Active and NAV<=2 and !FCLLM. Send_NAV_RGV)	True	UCA-9
PUCA-12	E()(Modes==1 and FCPLM. APPR_Button and LAPPR_Track_Cond==1 and !MLLAPPR. VAPPR_Active and LAPPR<=2 and !FCLLM. Send_LAPPR_RGV)	True	UCA-10
PUCA-13	E()(Modes==1 and FCPLM. GA_Button and !MLLGA. LGA_Selected and LGA<=1 and !FCLLM. Send_LGA_RGV)	True	UCA-11
PUCA-14	E()(Modes==1 and ADS. OverSpeed and MLLGA. LGA_Selected and LGA==2 and FCLLM. Send_LGA_PGV)	False	—
PUCA-15	E()(Modes==1 and FCPLM. HDG_Button and MLNAV. NAV_Active and NAV==3 and FCLLM. Send_NAV_RGV)	True	UCA12

通过验证可以发现，PUCA-7，PUCA-9 和 PUCA-14 在可达性验证中不通过，所以在现存的系统设计中不存在任何一种情况使得这两种 PUCA 发生，故暂不能认定为 UCA。

4.4.5 致因因素和损失场景分析

本文方法最后一步是进行致因因素和损失场景分析，这一步需要结合大量的专业领域知识，在 STPA 方法中致因因素分析框架的帮助下，逐步排查可能导致 UCA 产生的原因以便针对性地进行安全约束。致因因素分析框架如图 16 所示。

本文以“UCA-1：FGS 在飞行员/飞行计划选择了 VS 模式以设定好的垂直速度进行爬升或下降时未提供 VS 模式相关的俯仰制导命令”为例展开进一步分析。

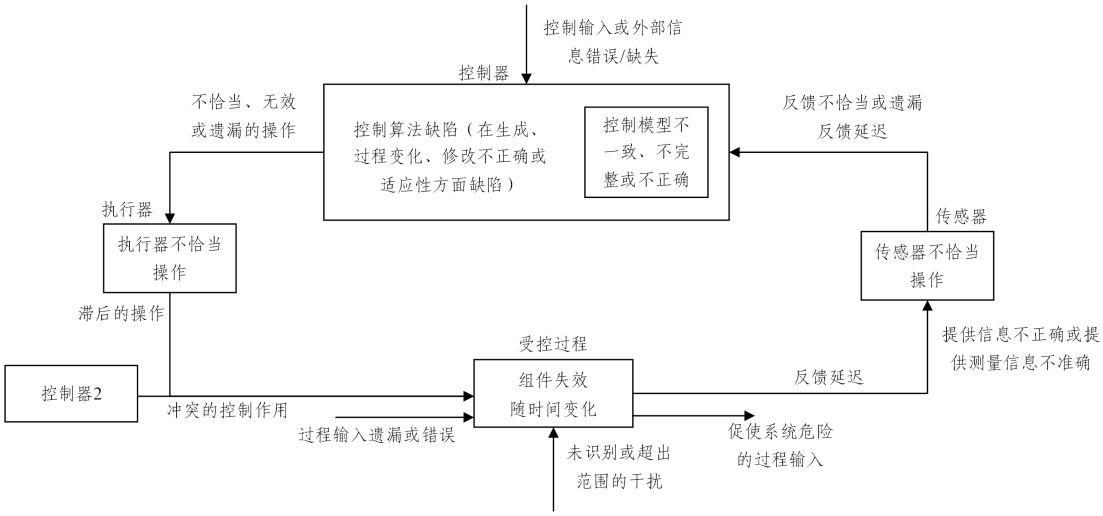


图 16 STPA 致因因素分析框架图

Fig. 16 Framwork diagram of STPA casual factor analysis

根据致因分析识别出的 UCA-1 致因因素与损失场景如下。

1）与不安全控制器行为相关的致因因素

模式逻辑在接收到飞行员发出的打开 VS 模式指令之后，还未来得及发送 VS 模式被激活的信号给飞行控制律模块，便接收到了来自 FMS 飞行计划所设定的激活其他非基本垂直模式的指令，造成了模式混乱。FGS 选择优先处理选中的飞行模式指令，而未发送 VS 模式相关的俯仰制导指令。

2）与控制路径相关的致因因素

FGS 内部顺利地处理了打开 VS 模式进行飞行的相关工作，但是由于 FGS 与 AP 之间的驱动线路发生故障，AP 无法接收到 FGS 发出的 VS 模式相关俯仰制导指令。

根据专家评判，上述两种情景均会导致 UCA-1 的发生。2009 年，法航 447 号航班在飞往巴西途中，飞机遭遇气象干扰，飞行员未能有效处理飞行导引系统生成的控制指令，导致空难的发生。事故报告指出[2]，发生该事故的其中一个原因是飞行员对飞行管理系统的反馈产生误判，导致未能及时纠正失速状态。飞行员的操控意图与 FMS 的飞行计划出现冲突，最终导致飞机坠毁于大西洋。而本节分析出的致因因素 1 能够覆盖该事故原因。

致因因素框架中展示了 15 类损失场景，而针对一个 UCA 进行损失场景与致因因素分析，需要逐一分析这些致因因素。示例中共分析了 15 个 PUCA，筛选后留下 12 个有发生概率的 UCA，节省了 45 次分析过程。数据表明，使用模型

检验对不安全控制行为进行筛选能够提高系统分析的效率，减少资源的浪费。具体数据如表 5 所列。

表 5 致因因素分析量对比

Table 5 Comparison of number of causal factors

安全性分析方法	UCA 识别		实际 UCA	致因因素分析量总计
	已识别	需筛除		
传统 STPA	15	0	15	225
扩展 STPA	15	3	12	180

结束语 本文针对飞行导引系统模式转换这类复杂系统的安全性问题，提出了一种将 STPA 与模型检验相结合的方法，并借助 NASA 简化的飞行模式案例进行了详细展示。首先，选用基于系统理论的 STPA 方法作为安全性分析的主要方法，避免了传统方法中存在的忽略组件交互因素问题；然后，引入基于时间自动机理论的形式化模型检验工具 UPPAAL 对 STPA 构建的模型以及识别出的 PUCA 进行检验，一定程度上解决了 STPA 由于人工分析产生的缺乏形式化的问题；最后，使用系统的致因因素分析框架对 UCA 进行逐一分析，并分析了扩展方法相较于传统 STPA 的致因因素分析量。数据表明，本文方法对提高资源利用率有所帮助。

然而，本文依然存在一些局限性。1）自动化程度不足。在使用 UPPAAL 对 PUCA 进行检验之前，PUCA 的识别是依靠领域专家的头脑风暴，不论是领域专家直接分析还是对枚举结果进行初筛，都难以避免地需要人工分析，降低了分析效率。致因因素的分析同样如此。未来可以针对这些非自动化分析部分进一步寻找优化方案。2）选用的 NASA 简化模型案例并未覆盖实际工业飞行导引系统的全部飞行模式，部分模式的转化条件也有所简化，对飞行导引系统模式转换领域而言，分析工作并不完整。未来还需在此基础上增加对其他模式的分析，完善飞行导引系统模式转换的安全性分析工作。

参 考 文 献

[1] SAVELEV A, NERETIN E. Development of safety requirements for tracking active pilot controls by signals from an automatic flightcontrol system[C]// Proceedings of the 2019 International Conference on Control, Artificial Intelligence, Robotics & Optimization(ICCAIRO). IEEE, 2019: 19-24.

[2] OLIVER N, CALVARD T, POTOČNIK K. Cognition, technology, and organizational limits: Lessons from the Air France 447 disaster[J]. Organization Science, 2017, 28(4): 729-743.

[3] BOSS K K. Characteristics and analysis of major US air carrier accidents between 1991 and 2010[M]. Oklahoma: Oklahoma State University, 2012.

[4] LI X, ZHU Y, FAN Y, et al. A Comparison of SAE ARP 4754A and ARP 4754[J]. Procedia Engineering, 2011, 17: 400-406.

[5] LEVESON N, FLEMING C, THOMAS J, et al. A Comparison of SAE ARP 4761 and STPA Safety Assessment Processes [C]// Safety-Critical Systems Symposium. 2015.

[6] QIE Z, YAN H. A Causation Analysis of Chinese Subway Construction Accidents Based on Fault Tree Analysis-Bayesian Network[J]. Frontiers in Psychology, 2022, 13: 887073.

[7] ANJALEE J A L, RUTTER V, SAMARANAYAKE N R. Application of failure mode and effect analysis(FMEA) to improve medication safety: a systematic review[J]. Postgraduate Medical Journal, 2021, 97(1145): 168-174.

[8] MILLER S P, CARLSON T M, TRIBBLE A C. Flight guidance system requirements specification[R]. NASA, 2003.

[9] TRIBBLE A C, LEMPIA D L, MILLER S P. Software safety analysis of a flight guidance system[C]// Proceedings of the 21st Digital Avionics Systems Conference. IEEE, 2002.

[10] JOSHI A, MILLER S P, HEIMDAHL M P E. Mode confusion analysis of a flight guidance system using formal methods[C]// Digital Avionics Systems Conference. 2003.

[11] OWRE S, RUSHBY J M, SHANKAR N. PVS: A prototype verification system[C]// International Conference on Automated Deduction. Berlin: Springer, 1992: 748-752.

[12] CIMATTI A, CLARKE E, GIUNCHIGLIA F, et al. NuSMV: A new symbolic model verifier[C]// Computer Aided Verification: 11th International Conference. Berlin: Springer, 1999: 495-499.

[13] WOODCOCK J, LARSEN P G, BICARREGUI J, et al. Formal methods: Practice and experience[J]. ACM Computing Surveys, 2009, 41(4): 1-36.

[14] SULAMAN S M, BEER A, FELDERER M, et al. Comparison of the FMEA and STPA safety analysis methods-a case study[J]. Software Quality Journal, 2019, 27: 349-387.

[15] LEVESON N G. Engineering a safer world: Systems thinkingapplied to safety[M]. Massachusetts: MIT Press, 2016.

[16] LEVESON N, DULAC N, ZIPKIN D, et al. Engineering resilience into safety-critical systems[M]// Resilience Engineering. 2017: 95-123.

[17] CHAAL M, BANDA O A V, GLOMSRUD J A, et al. A framework to model the STPA hierarchical control structure of an autonomous ship[J]. Safety Science, 2020, 132: 104939.

[18] SAHAY R, ESTAY D A S, MENG W, et al. A comparative risk analysis on CyberShip system with STPA-Sec, STRIDE and CORAS[J]. Computers & Security, 2023, 128: 103179.

[19] SADEGHI R, GOERLANDT F. A proposed validation framework for the system theoretic process analysis (STPA) technique[J]. Safety Science, 2023, 162: 106080.

[20] BENSACI C, ZENNIR Y, POMORSKI D, et al. STPA and Bowtie risk analysis study for centralized and hierarchical control architectures comparison [J]. Alexandria Engineering Journal, 2020, 59(5): 3799-3816.

[21] BENSACI C, ZENNIR Y, POMORSKI D, et al. Collision hazard modeling and analysis in a multi-mobile robots system transportation task with STPA and SPN[J]. Reliability Engineering & System Safety, 2023, 234: 109138.

[22] TSUJI M, TAKAI T, KAKIMOTO K, et al. Prioritizing scenarios based on STAMP/STPA using statistical model checking

[C]// 2020 IEEE International Conference on Software Testing, Verification and Validation Workshops (ICSTW). IEEE, 2020: 124-132.

[23] DE SOUZA F, DE MELO B J, HIRATA C M, et al. Combining STPA with SysML modeling[C]// 2020 IEEE International Systems Conference (SysCon). IEEE, 2020: 1-8.

[24] ZHONG D, SUN R, GONG H, et al. System-theoretic process analysis based on SysML/MARTE and NuSMV[J]. Applied Sciences, 2022, 12(3): 1671.

[25] DGHAYM D, HOANG T S, TURNOCK S R, et al. An STPA-based formal composition framework for trustworthy autonomous maritime systems[J]. Safety Science, 2021, 136: 105139.

[26] BEHRMANN G, DAVID A, LARSEN K G. A tutorial on uppaal[EB/OL]. https://uppaal.org/texts/21-tutorial.pdf.

[27] LIU S, ZHOU C, SHAO H, et al. Research on automatic flight control system flight mode operation logic[C]// Journal of Physics: Conference Series. 2023.

[28] HÜBENER D, LUCKNER R, WEBER G. Concepts for Independent Monitoring of Flight Control Laws[C]// Proceedings of the 10th EUCASS-9th CEAS Aerospace Europe Conference 2023. 2023.

[29] HANDBOOK A F. Federal Aviation Administration[R]. Department of Transportation, 1972.

[30] ISHIMATSU T, LEVESON N G, THOMAS J, et al. Modeling and Hazard Analysis Using STPA[C]// Proceedings of the 4th IAASS Conference. 2010.

[31] MCCRACKEN D D, REILLY E D. Backus-naur form (BNF)[M]// Encyclopedia of Computer Science. 2003: 129-131.

[32] MAIDI M. The common fragment of CTL and LTL[C]// Proceedings of 41st Annual Symposium on Foundations of Computer Science. IEEE, 2000: 643-652.

ZUO Chencui, born in 2000, postgraduate. Her main research interests include formal methods and system safety analysis.

HUANG Zhiqiu, born in 1965, Ph. D, professor, is a distinguished member of CCF (No. 09028D). His main research interests include software engineering, safety of software and formal methods.

(责任编辑:何杨)