使用系统理论过程分析对自适应系统进行危险分析

Simon Diemert计算机科学系 维多利亚大学维多利亚，加拿大 sdiemert@uvic.ca

JensH. Weber计算机科学系 维多利亚大学维多利亚，加拿大 jens@uvic.ca

摘要——**自适应系统能够在运行时响应变化而改变其行为。自适应是管理现代系统设计过程中存在的不确定性的重要策略，例如自动驾驶车辆。然而，确保自适应系统的安全仍然是一项挑战，尤其是在自适应调整影响安全关键功能时。安全工程领域已建立了分析系统安全性的实践方法。系统理论过程与分析（STPA）是一种非常适用于自适应系统的危险分析方法。本文介绍了针对自适应系统的 STPA 设计时扩展。随后，本文推导出一个参考模型和分析义务，以支持 STPA 活动。该方法应用于文献中描述的三个自适应系统。结果表明，按照文中所述方式使用 STPA，是一种适用于安全关键型自适应系统的危险分析方法。**

索引词—**安全关键系统、自适应系统、危险分析、系统理论事故模型过程、系统理论过程分析**

I. 引言

现代系统越来越需要应对因运行条件多变且不断演变而产生的不确定性。适应是应对这种不确定性的重要机制。能够响应自身或环境的变化而改变其行为或结构的系统称为“自适应系统”（SAS） [1]。适应已应用于广泛领域，包括 Web 服务管理 [2]–[4]，、无线传感器网络 [1]，[5]–[7]，以及机器人学和自主系统 [8]–[13]。

许多应用领域中，自适应系统可能导致事故，即造成人员、环境或财产受到伤害的损失事件 [14]。例如，自动驾驶地面车辆中的自适应功能可能错误地调整电子制动控制器的特性，导致制动效果不如当前条件所要求的那样有效。执行安全关键功能的自适应系统称为“安全关键自适应系统” [15]。我们需要开发方法，以确保用于高风险应用领域的自适应系统 [16]。事实上，针对安全性以及可靠性或效率等其他质量属性对自适应系统进行保障，已成为众多研究项目的主题 [13]，[17]–[19]。

已有成熟的工程实践可用于确保传统（非自适应）系统的安全。这些实践的描述见于标准之中，这些标准规定了工程过程模型，以增强人们对系统达到可接受安全水平的信心 [20]–[23]。危险分析是一项重要活动，包括识别危险、评估其关键性，并分析设计以确定危险原因。故障树分析（FTA）、失效模式与影响分析（FMEA）和危险与可操作性分析（HAZOP）等方法均用于此目的 [20]，[23]，[24]。确定危险原因后，标准要求实施安全需求（即“缓解措施”）以降低风险。

当前最先进的自适应系统危险分析方法，要么仅涉及设计时方法 [25]–[28] ，要么采用混合方法（设计时和运行时方法） [17]，[29]–[31]。混合方法使用传统的设计时分析（例如故障模式与影响分析）识别危险原因，然后采用各种方法评估系统在运行时是否容易受到这些原因的影响。现有方法存在三方面的局限。首先，它们采用的危险分析方法对事故因果关系的理解较为有限 [14]。具体而言，这些方法假设危险是由技术组件故障（例如阀门卡滞、内存损坏、短路等）导致的。这一假设在这些方法最初创建时适用于简单系统；然而，对于更复杂的系统（如自适应系统）则并不适用，因为危险也可能由组件之间的交互作用引起 [14]，[32]，[33]。其次，现有方法关注被管理系统中产生的危险原因（被管理系统执行系统功能）；它们没有考虑管理系统（执行自适应调整）可能促成危险发生的可能性。然而，由于管理系统通常是复杂的软件，因此它们也可能发生故障，进而影响被管理系统的安全相关功能 [16]。第三，尽管现有方法使用传统分析方法（如故障模式与影响分析）的输出，但它们并未明确将自适应系统特定方面作为产生危险原因的设计时活动的一部分加以考虑。通常会使用系统或技术特定的参考模型或引导短语来辅助专家开展传统危险分析。然而，目前尚无针对自适应系统的专门指导，因此某些危险原因可能会被忽略。

总而言之，为使自适应系统获得利益相关者和监管机构的认可并用于安全关键应用，有必要

证明整个自适应系统（包括管理系统）均已接受系统性的危险分析，并且该分析考虑了 SAS 特定方面。因此，需要采用整体性事故因果关系概念的危险分析方法，将被管理系统的行为可能促成危害发生这一情况纳入考量。

A. 贡献——面向自适应系统的系统理论过程分析 *tems*

尽管在运行时开展危险分析是一项重要的研究方向，[34]，本文所述工作侧重于推进设计时危险分析，采用一种系统理论方法，并融入自适应系统参考架构的细节。

系统理论过程分析（STPA）是一种相对较新的危险分析方法（以安全工程领域的发展历程为尺度），由 Leveson 提出 [14], [32]。STPA 的一个核心思想是，危害不仅源于组件故障，也可能源于系统组件之间的正常交互，其中还包括组件或环境发生变化（适应或演化）的可能性。STPA 采用一种称为系统理论事故模型与过程（STAMP）的系统建模方法，通过反馈控制回路描述系统组件之间的关系；这与反馈控制以及知识库监视器-分析-计划-执行（MAPE-K）范式的理念相似，该范式已广泛应用于自适应系统 [35], [36]。鉴于二者在概念上的相似性，STPA 非常适合用于自适应系统的危险分析。

尽管系统理论过程分析与自适应系统之间具有协同效应，且系统理论过程分析相较于其他危险分析方法具有优势 [14], [37]–[38], ，系统理论过程分析归根结底是一种通用方法。若不理解如何使用系统理论事故模型与过程对自适应系统建模，或不了解自适应系统达到“安全”意味着什么，就盲目应用系统理论过程分析，可能导致分析不充分。因此，最好将 Leveson 的通用系统理论过程分析方法针对自适应系统进行定制，正如该方法已针对其他应用领域进行定制一样 [39]–[42]。

本文根据系统所执行的自适应调整类型，将 Leveson 最初提出的系统理论过程分析扩展至自适应系统。扩展后的方法在设计时应用，以系统化地识别源自受控系统和管理系统的危害原因。除了发现危害原因，该方法还为受控系统和管理系统识别安全需求，以降低风险。作为扩展方法的一部分，本文推导出适用于自适应系统的参考 STAMP 控制结构，并说明了分析义务，即执行系统理论过程分析时需要考虑的条件。本文将扩展后的方法应用于三个自适应系统，以评估其效果。据我们所知，这是首篇将系统理论过程分析应用于自适应系统的论文。

II. 基础

根据 Weyns 的定义，自适应系统（SAS）是符合两个原则的系统 [1]。外部原则规定，自适应系统必须能够自主处理来自环境的变化和不确定性，在自身内部完成处理，并实现其目标。内部原则规定，自适应系统由两个主要部分组成：一个与环境交互，另一个负责处理适应相关事务，并使用反馈回路与前一部分进行交互。自适应系统中与环境交互的部分称为 被管理系统，而系统中负责处理适应的部分称为管理系统 [36]。适应目标描述了需要满足的高层目标，并被提供给被管理系统。MAPE-K参考模型被广泛用于实现管理系统的适应能力 [35], [36] ，也是第 III 节中 STAMP参考控制结构的基础。

A. 安全关键自适应系统

先前的研究已为安全关键自适应系统确立了定义和概念，这些内容为本文提供了依据 [15], ，现简要重述如下。

危害是系统安全工程活动的基础概念，也是系统与其运行环境之间的概念性“桥梁”；它们是最高层级的事件，其他所有安全工程活动都由此展开。更正式地说，危害 是指“系统的一种状态或一组条件，当与一组特定的最坏情况环境条件共同作用时，将导致事故（损失）” [14]。安全关键系统 是指这样一种系统：其错误的、无意的或顺序错乱的操作，与环境中的条件相结合，会导致危害发生（或未能减轻危害的影响） [23], [43]。若某系统满足以下条件，则称其为安全关键自适应系统：i）满足外部原则；ii）满足内部原则；iii）被管理系统属于安全关键系统 [15]。

“适应选项”是管理系统生成的被管理系统候选配置，用于满足适应目标 [1]。“安全适应选项”是指“应用于被管理系统时，不会导致或促使被管理系统进入危险状态”的适应选项 [15]。可以通过设计时或运行时分析活动（如测试、仿真和形式化验证）证明某个适应选项是安全的。通常，这些活动的结果会记录在论证案例中，并可能随着系统适应而动态更新，纳入新的保证证据 [16], [18], [44], [45]。

自适应系统执行的自适应调整可根据其对被管理系统安全关键功能的影响（假设自适应系统正常运行）进行分类 [15]。对自适应调整进行分类，是第III节所述扩展 STPA方法中的重要步骤。自适应调整分为四种类型：

1）类型 *0* - 非干扰：这些自适应调整不会影响被管理系统的安全关键功能。

2）*I* 型 - 静态保证：这些自适应调整（以及下文的 II 型和 III 型自适应调整）会影响被管理系统的安全关键功能。管理系统只从预先确定的一组安全选项中选择适应选项。每个选项在设计时均已证明是安全的，无论系统可能遇到何种运行条件。

3) II 型 - 受约束保证：与 I 型一样，管理系统从预先确定的

集合中选择适应选项。然而，其安全性取决于环境或系统的状态。在应用某个选项之前，管理系统会更新动态保证论证，并加入证据以证明该选项的安全条件已得到满足。一个特殊情况（IIb 型）是，每次后续适应都会单调地增强对环境或系统状态的假设（即变得更具限制性）。

4) III 型——动态保证：在设计阶段未定义适应选项，或无法证明其安全性。必须在运行时评估适应选项，并将评估结果作为适应操作的一部分纳入动态安全论证。

在此前关于 [15]，II 型适应的研究中，我们还要求后续适应中的约束单调递增。然而，本文采用更一般的定义：只要某一配置的运行条件得到满足，系统就可以返回到先前的配置。

B. 系统理论事故模型与过程和系统理论过程分析

Leveson 主张采用整体性的事故因果关系观，认为“事故源于对安全相关行为的约束控制或执行不足。……”[14, 第 100]页。由此可见，安全是一个控制问题，而非组件可靠性问题。从这一视角来看，STAMP 提供了一种基于反馈和控制流的事故致因模型。STAMP 模型是一种“控制结构”，描述系统各要素之间的功能（控制）关系。

图中展示了一个简单的 STAMP 控制结构。1。STAMP 控制结构中的元素包括：受控的过程、提供控制动作的控制器、观测状态的传感器，以及执行控制动作的执行器。控制器还可能具有一个内部过程模型，用于模拟其交互元素的状态和行为；过程模型只是对现实的不完全反映，因此可能不准确或有误。控制结构可以按层级嵌套，以反映现实世界中系统之间的关系。系统理论过程分析是一种基于 STAMP 因果模型的危险分析方法。值得注意的是，系统理论过程分析有助于分析人员考虑组件失效之外的因果因素，包括控制冲突与交互、人类决策和认知错误、社会/组织因素以及系统性设计缺陷（例如软件缺陷） [14]。系统理论过程分析的总体目标有两个：识别危害可能发生的方式，然后确定能够防止危害发生的缓解措施。系统理论过程分析分为四个步骤 [46]:

1) 识别损失和危险：通过识别不可接受的损失事件，了解当前的“安全问题”。然后识别可能导致损失的危险状况。

2) 建立控制结构模型：采用STAMP建模范式，并选择适当的抽象层次，将系统描述为一个控制结构。

3）识别不安全控制行为：使用引导短语（“已提供”、“未提供”、“过晚/过早”、“持续过久/过早停止”），

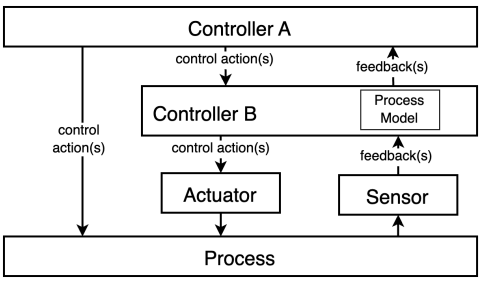


图1. 简单的STPA控制结构。

分析控制结构中的每个控制动作，以判断它是否可能导致危险发生，即是否属于不安全控制动作（UCA）。记录一条简短、格式规范的语句来描述每个 UCA。

4) 识别损失/因果场景：针对每个不安全控制动作，识别一个或多个场景，说明该动作可能如何导致危险发生。在此步骤中，分析人员会考虑造成问题的控制器及其他要素的因果因素。可参考一张通用控制结构图，图中标注了因果因素的提示。

可在步骤3或步骤4之后识别缓解措施。在某些情况下，可立即采取措施缓解不安全控制动作；在其他情况下，则需要进一步分析以了解所需的缓解措施。

STAMP和STPA已发展出若干变体和扩展方法：事故分析系统理论是一种互补方法，用于事后确定事故是如何发生的 [14]；Thomas将STPA的某些方面形式化，以生成安全需求 [47]；Young和Leveson扩展了该方法，将网络安全纳入其中 [39]；Stringfellow和France均扩展了该方法，将人为和组织因素纳入其中 [48], [49]；Weber和Mason-Blakely则将STAMP扩展应用于电子病历 [41]。

III. 自适应系统的STPA

自适应系统通常可用反馈控制回路来描述，因此适合使用STAMP建模，并通过STPA进行分析。本节通过以下方式调整现有的STPA方法，使其适用于自适应系统：1）增加分析步骤，以识别并分类自适应调整；2）推导出适用于自适应系统的专用参考STAMP控制结构；3）确定可用于指导分析的分析义务。

A. 自适应系统的扩展*STPA*方法

图2 展示了面向自适应系统的扩展STPA方法概览。分析步骤以圆角矩形和粗体文字表示，分析步骤之间的推进过程（“控制流”）则以实线反向箭头表示。传统STPA的步骤以黑白框表示，而针对自适应系统的扩展则以蓝色阴影框和箭头表示。传统STPA分析方法已在第II节中介绍；下文将详细讨论每个新增要素。

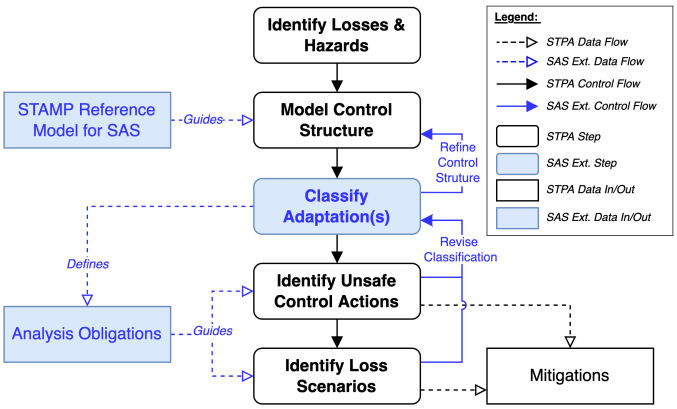


图 2. 自适应系统扩展STPA方法概述。

1）*STAMP*参考模型：STAMP 控制结构的建模以我们面向自适应系统的STAMP参考模型为指导（参见III-B节）参考模型描述分析人员可纳入特定系统建模中的建模模式。针对STPA其他定制版本的STAMP参考模型也已开发；例如，Mason-Blakey和Weber描述了一种参考模型，该模型将STAMP定制为包含电子病历的医疗系统 [41], [42]。

2) 适应分类与分析义务：在对控制结构建模后，应识别并分类自适应系统执行的适应。根据安全关键型适应的分类体系 [15]，将适应分为四种“类型”（类型 0、I、II、III）。适应类型用于确定分析义务，以指导识别不安全控制行为和损失场景。

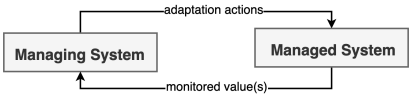
STAMP 控制结构的一项优势是，可以根据分析需要调整模型的详细程度：建模后的控制结构应包含足够的细节，使分析人员能够推理适应及其对安全的影响。可根据需要通过迭代完善来增加细节。此外，在识别不安全控制行为或损失场景时，分析人员可能会重新考虑某项适应的分类。因此，图中还展示了一个修订反馈步骤。2。最后，尽管图中未予展示（因为这属于原始STPA方法的一部分），分析人员也可以选择在识别不安全控制动作或损失场景时完善控制结构。

B. 面向自适应系统的 *STAMP*参考模型

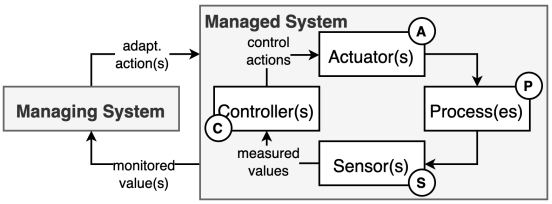
本节采用一系列建模步骤，逐步推导出适用于通用自适应系统的 STAMP 控制结构。推导过程（以及我们的论证依据）以非正式的构造性证明形式呈现，旨在说明图3所示最终参考模型的有效性。建模步骤也可以应用于实际系统，以得出相应的控制结构。

在系统图中，组件以无阴影矩形表示，并标注一个“徽章”以指示其元素类型。例如，控制器类型的元素标注为“C”，探针类型的组件标注为“Pb”。阴影框用于划分被管理系统与管理系统。

1) 初始模型：推导从被管理系统和管理系统开始。管理系统对被管理系统执行自适应动作，并监视来自被管理系统的值。在为实际系统推导模型时，应在此步骤中描述具体的自适应动作和受监视变量。



2) 细化被管理系统：对被管理系统进行细化，使其至少包含一个过程，并可选择性地包含一个控制器。根据系统的不同，可以对多个控制器和过程进行建模。过程既可以表示受控量，也可以表示环境的某些方面。控制结构中还可以选择性地包含多个传感器和执行器组件，用于被管理系统。根据所需的详细程度，可能不必包含传感器和执行器：即使没有它们，也可以执行系统理论过程分析，而且许多已发表的示例中也未对其进行建模 [14]。

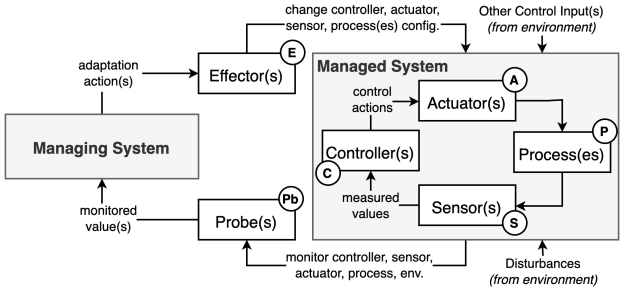


3) 添加来自环境的外部控制输入和扰动：向被管理系统添加其他控制输入和外部扰动。这些输入和扰动源自未建模的控制器或过程，或来自环境，并可能影响被管理系统中的元素。如果输入和扰动与分析无关，则无需执行此步骤。

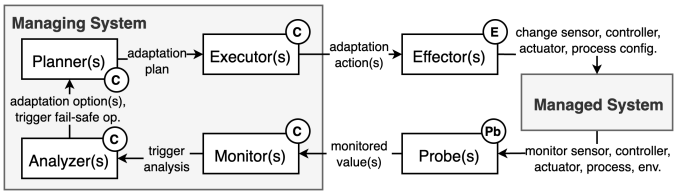
4) 添加探针和执行器：探针和执行器构成管理系统与被管理系统之间的接口。探针负责监视特定方面，并向管理系统提供数据。执行器根据适应操作改变被管理系统的配置。至少应包含一个探针和一个执行器。每个控制器、执行器或传感器都可以连接到探针或执行器。允许探针和执行器连接到控制器的理由显而易见。传感器提供过程的测量数据，而这些数据可能是管理系统所必需的，因此可以将传感器连接到探针。管理系统可能会改变传感器的行为（例如，调整其灵敏度），因此也可以将传感器连接到执行器。对于执行器，管理系统可能会改变其行为（例如，启用或禁用执行器），因此可以将其连接到执行器。此外，管理系统可能需要

监视执行器的属性（例如，物理性能），因此可以将其连接到探针。根据 Weyns 的说法，探针还可用于直接监视被管理系统内的过程或环境 [1]。执行器可以直接更改过程，但无法直接影响环境。

探针和执行器分别类似于传感器和执行器。实际上，可以将探针视为传感器的一种（反之亦然），也可以将执行器视为执行器的一种。我们将它们分开，以明确其角色：探针和执行器连接管理系统与被管理系统，而执行器和传感器存在于被管理系统内部，位于控制器与过程之间。



5) 细化管理系统：在第五步中，对管理系统进行细化，使其包含监视器、分析器、规划器和执行器组件。从控制理论视角来看，这些组件中的每一个都被视为控制器的一种，向其他元素提供控制动作。根据所建模系统的具体情况，此步骤允许存在多个监视器、分析器、规划器或执行器。在下图中，被管理系统的详细内容被“折叠”为一个单独的模块，以节省空间。



6) 添加知识库：知识库是管理系统中所使用的 MAPE-K 参考模型的重要组成部分。在第六步中，将知识库纳入控制结构。结果如下图 3 所示。MAPE 组件与知识库之间的关系基于 Weyns 对 MAPE-K 的描述 [1]。在通用的 STAMP 控制结构中，控制器可以选择包含一个“过程模型”，用于描述控制器对其所控制的过程的内部表征。管理系统中的知识库具有类似用途，因此被视为共享过程模型。Weyns 描述了知识库中可能包含的四种模型：被管理系统模型、环境模型、目标模型和工作模型[1]。被管理系统模型表示真实被管理系统的状态。环境模型描述环境的状态。适应目标模型描述自适应系统的目标，并用于生成和分析适应选项。最后，工作模型用于执行特定的分析任务，例如预测某个适应选项对被管理系统的影响。即使在实际系统中没有明确纳入这些模型，在此参考控制结构中纳入它们，也有助于提醒人们知识库中可能存储的数据。

7) 纳入高阶适应：其他控制元素可能会影响管理系统，例如改变适应目标 [1]。这种“高阶”适应可能与自适应系统有关。实际上，Leveson 和 Stringfellow 都使用高阶控制器来描述影响“主要”控制器行为的人类或技术决策 [14], [48]。在最后一步中，可以选择性地纳入这些高阶控制元素，并将其连接到管理系统。最终的参考模型如图3所示。

C. 按适应类型划分的分析义务

建立 STAMP 控制结构模型并确定适应类型后，系统理论过程分析便会着手识别不安全控制行为和损失场景。根据系统类型，本节所述的“义务”为分析提供指导。下文以叙述形式说明这些义务，并在图4中将其概括为因果因素。为简洁起见，这些义务主要针对适应、管理系统，以及被管理系统与管理系统之间的关系；其他（非适应相关）分析义务也可能适用。

1）Type0 适应：根据定义，Type0 适应不会干扰系统的安全相关功能，因此其正常行为不太可能导致危险发生。不过，在识别损失场景时，分析人员应考虑执行器或探针运行不充分或不正确，进而干扰被管理系统的可能性。例如，假设某个探针为了测量控制器的属性而发送过多网络请求，以致控制器中的关键任务得不到足够的处理器时间。

*2) I*型适应：I型适应的适应选项通常从预先确定的一组选定安全选项中选取。然而，管理系统可能因缺陷而提供无效的适应选项（即不属于已知安全选项集）。因此，在识别不安全控制行为时，分析人员应考虑以下可能性：分析器、规划器或执行器生成、选择或应用了不属于预先确定的安全适应选项集的适应选项。

在识别损失场景时，分析人员应考虑执行器可能会错误地执行某项自适应动作，从而导致产生不安全配置。此外，由于探针可能会测量被管理系统的安全相关属性，分析还应考虑探针对被管理系统的安全相关功能造成干扰的可能性。最后，分析还应考虑适应选项集合是否

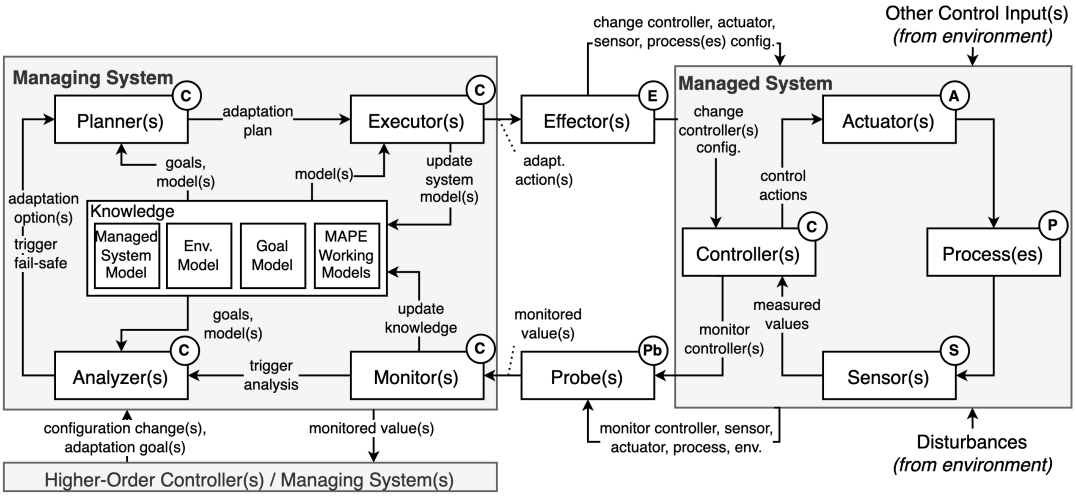


图 3. 自适应系统的 STAMP 参考模型。

与预先确定的安全选项集合不同，例如由于数据损坏所致。

3) II 型自适应：II 型自适应在设计时被证明是安全的，但需满足针对被管理系统或环境的一组约束。如果这些约束得到满足，则被管理系统可确保安全。识别不安全控制行为时，应考虑分析器、规划器或执行器应用不安全的自适应选项的可能性，包括在违反其安全约束时仍应用某个自适应选项的可能性。此外，由于选择安全适应选项取决于运行环境或被管理系统的状态，因此还应考虑以下可能性：监视器未能或未准确更新知识库、更新延迟、遗漏对知识库的更新，或不必要地触发分析。例如，如果监视器未能更新知识库中的环境信息，分析器可能会错误地认为某些环境约束已得到满足，从而选择不安全的自适应选项。最后，下一个（安全）适应选项的选择可能取决于当前的系统配置，因此分析应考虑以下可能性：执行器在应用自适应动作时遗漏、延迟或错误地更新系统模型。

与 I 型相同，在识别 II 型自适应的损失场景时，分析应考虑执行器运行不充分或不正确可能对被管理系统造成的影响。此外，分析还应考虑，运行不充分可能导致探针向管理系统提供延迟、不准确、错误或缺失的数据。最后，知识库中的模型可能不完整、不正确或彼此不一致。

4) III 型自适应：III 型自适应的分析义务与 II 型相似，但还需考虑以下情况。识别不安全控制行为时，分析应考虑分析器延迟，或未触发故障安全控制安全动作，以及规划器或执行器延迟提供或未提供相应故障安全系统配置的可能性。

同样，识别损失场景时，分析应考虑执行器延迟应用故障安全配置变更或未应用这些变更的可能性。最后，分析应考虑知识库中的适应目标已过时或相互冲突，导致分析器提供不安全的自适应选项的可能性。

IV. 评估

面向自适应系统的 STPA 方法从三个方面进行评估：适用性、可重复性和有效性。由于本文是首次将 STPA 应用于自适应系统，评估重点在于适用性，并对可重复性和有效性作初步探讨。

A. 适用性

为评估适用性，我们将该方法应用于两个示例自适应系统以及我们自己的玩具热水器系统。除热水器系统外，[15], 这两个示例系统选自一份公开列表，筛选依据包括它们在自适应系统领域的知名度、应用的安全相关性、管理系统是否采用 MAPE-K 参考模型，以及是否有已发表的系统描述。选定的系统为：Delta IoT 无线传感器网络 [1], [7], [50] 和 UNDERSEA 无人水下航行器（UUV）[10], [18], [51]。将该方法应用于这些示例系统的结果汇总于表 1，其中列出了各系统的自适应类型，以及识别出的危害、不安全控制行为、损失场景和缓解措施的数量。下文将介绍该方法在这些系统上的应用，并进一步讨论其适用性。

1) 热水器的系统理论过程分析（*STPA*）：热水器系统存在一种危害：*H1* - 温度过高。由于热水器采用基本的反馈控制器和 MAPE-K 架构，其控制结构与图中参考模型几乎完全相同。1，因此为简洁起见，此处不再重复。热水器的适应属于 II 型

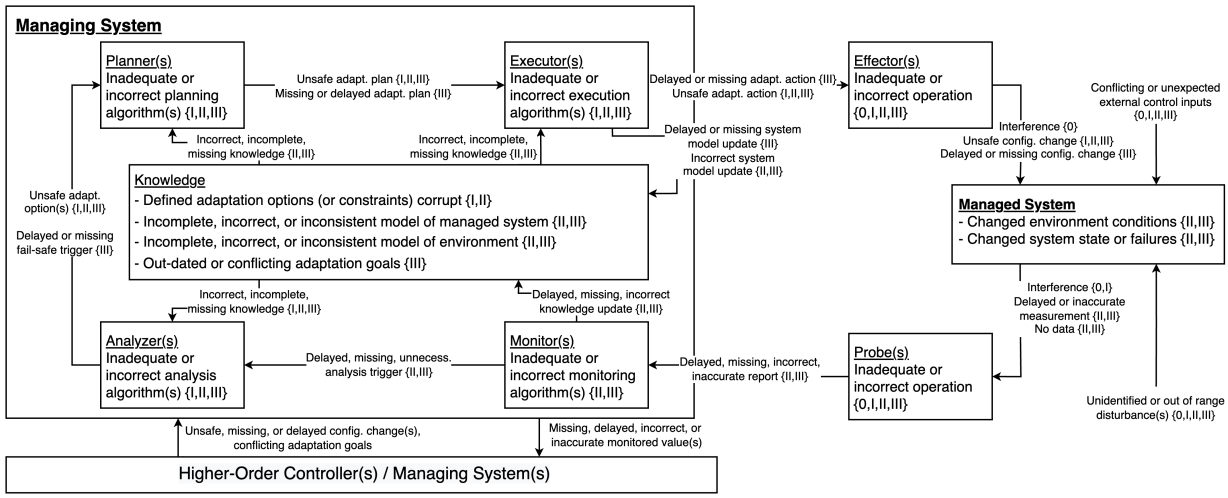


图 4. 因　识别损失场景时需考虑的因果因素汇总；适应类型如 {...} 花括号所示。

表 I 各系统分析结果汇总。

系统	Type	Hzd.	UCA	场景。	Mit.
热水器 [15]	II	1	8	12	15
Delta IoT [7]	III	2	29	33	48
UNDERSEA UUV [10]	III	3	45	60	50

因为管理系统从预先确定的一组控制器增益中进行选择，而这些增益的安全性取决于当前环境条件。共识别出 8 种不安全控制行为。例如：监视器未能及时更新输入流知识，导致分析器在分析环境条件时无法获知最新数据；如果据此使用错误的条件来选择下一组控制器增益，可能会导致温度过高事件。此外，还识别出 12 个损失场景和 15 项缓解措施。

2) Delta IoT 的系统理论过程分析：我们为 Delta IoT 系统识别出两项危害：*H1* - 未向安全人员报告和*H2* - 向安全人员报告的信息不准确。使用第III-B 节中的建模步骤，我们为 Delta IoT 制定了控制结构（见图5顶部）。我们没有对网络中所有可能的节点进行建模，而是使用单数形式的 Mote，它从受监视设施中的传感器读取数据，并向子节点传播数据和配置，或从子节点接收数据和配置。我们将 Delta IoT 执行的自适应调整归类为 III 型，因为这些调整会改变与安全相关的属性（例如传输属性），且系统没有可供选择的预定义配置集。系统共识别出 29 个不安全控制行为；例如：当网络性能下降到不可接受的程度或某个节点发生故障时，规划器未提供适应计划，从而导致危险H1发生。根据 III 型系统的分析义务（见图4），我们识别出 33 个损失场景；例如：

干扰会降低网络性能，导致部分节点无法传输数据包 *[*发生了变化的环境*]*；规划器从分析器接收适应选项；知识库中的适应目标相互冲突 *[*冲突目标*]*；规划器无法选择满足所有目标的配置，也没有向执行器输出任何计划 *[*计划缺失*]*；网络配置未发生变化 *[*动作缺失、配置变更缺失*]*；部分节点无法向其父节点传输数据 *[*控制动作缺失*]*；发生了事件，但未通知安全人员 *[H1]*。

参考模型中的因果因素用方括号表示 [...]。我们确定了 48 项缓解措施，并将其表述为应当陈述，Delta IoT 可实施这些措施，以缓解已识别的不安全控制行为和损失场景。

3) UNDERSEA UUV 的系统理论过程分析：我们为 UNDERSEA UUV 确定了三项危害：*H1* - 与地形失去间隔、*H2* - 数据缺失和*H3* - 数据不准确。应用第 III-B 节中的建模步骤后，得到了控制结构（见图 5底部）。UNDERSEA 执行的自适应调整属于 III 型，因为它们会影响系统的安全相关方面，而且适应选项是在运行时生成并经过验证（使用概率模型检验）的。我们确定了 45 项不安全控制行为，例如：当某个传感器的采集速率发生变化时，分析器提供的候选配置将无人水下航行器速度设得过高，超出传感器所能适应的范围，这可能导致危害 H3 发生。根据 III 型系统的义务，我们确定了 60 个损失场景，例如：

传感器的速率发生退化，探针将此情况报告给监视器；但是，知识库中关于传感器精度与速度的模型不准确 *[*错误知识*]*；分析器利用该模型选择了不兼容的传感器配置和速度*[*不安全的自适应选项*]*；新配置被应用后，*UUV*运动控制器提高了速度 *[*不当控制动作*]*；传感器的

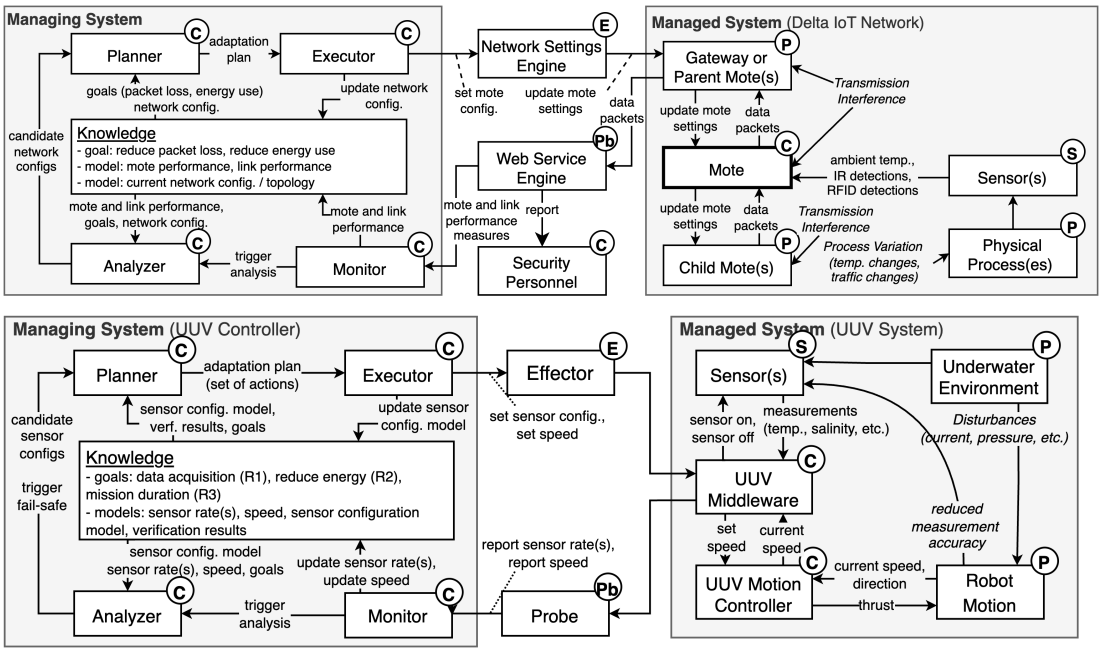


图5. 示例系统的控制结构；上：Delta IoT [7]，下：UNDERSEA UUV [10]。

精度随着速度的提高而下降，并记录了不准确的测量结果 *[H3]*。

总共确定了50项缓解措施，UNDERSEA 可通过实施这些措施来应对不安全控制行为和损失场景。

4）适用性观察：我们针对该方法的适用性提出三点观察。首先，应用第III-B 节中的建模步骤和 STAMP 参考模型，得到了细节程度适当的控制结构。这些控制结构并非过于简单，否则可能导致遗漏不安全控制动作或损失场景；同时也没有复杂到难以管理，否则可能产生不可信或难以理解的不安全控制动作或损失场景。此外，对于原作者所描述的系统组件、功能或行为，我们没有发现任何无法在所选抽象层次上纳入控制结构的内容。

其次，确定的不安全控制行为数量与其他已发表的、将系统理论过程分析应用于（非自适应系统）系统的研究结果相当 [42], [52]–[55]；不过，这一数量略低于对真实系统进行分析时的结果。这是因为我们的分析聚焦于系统的自适应方面（MAPE-K 等），而非被管理系统的细节。至于损失场景，由于可能存在组合数量众多的场景，分析仅选择确定一两个场景，用以说明不安全控制行为的背景，并展示该方法能够产生可信的损失场景。

第三，所涵盖的示例涉及多种系统和适应类型。示例包括无线传感器网络、无人机器人系统和基本反馈控制器。这表明该方法适用于广泛的自适应系统。此外，所考虑的适应类型为 II 和 III，它们（在很大程度上）也涵盖了类型 0 和 I 的适应义务。

B. 可重复性

对于危险分析方法，可重复性是指不同分析人员使用同一方法分析同一系统时，是否会得出相同结果 [56]。评估可重复性的一种方法是开展研究，让多位分析人员分别独立应用该方法，然后比较结果，例如，识别出的不安全控制行为和损失场景是否相似？此类研究超出了本文的范围。相反，我们针对可重复性提出两点观察。

首先，控制结构建模会因分析人员的视角不同而存在差异。由于控制结构是系统理论过程分析的基础，因此会对结果产生重大影响。第III-B 节中的推导步骤，最初旨在为我们的参考控制结构提供一种构造性证明。然而，在评估过程中，我们发现这些推导步骤有助于指导建模。具体而言，这些步骤帮助我们梳理被管理系统与管理系统之间关系的思路，并常常促使我们重新查阅源材料以确认细节。这一意料之外但并非不受欢迎的观察表明，对于采用 MAPE-K参考模型的自适应系统，建模步骤能够提升系统理论过程分析的可重复性。

其次，第 III-C 节中的分析义务通过确保至少考虑一组最小的不安全控制行为和因果因素，有助于提高可重复性。在评估过程中，我们发现这些义务有助于指导我们的思考，并促使我们重新考虑一些 SAS 特定方面；如果没有这些义务，这些方面可能并不明显。

C. 有效性

总的来说，很难评估一种危险分析方法是否能够有效地发现某项危害的所有原因。通过比较分析方法所识别的原因与系统整个生命周期中报告的危险发生情况，开展纵向研究或回顾性研究，是评估有效性的一种方式。然而，这类研究在实际开展中存在困难。因此，研究人员通常会在方法之间进行比较，以确定哪种方法能够产生更有意义的结果。关于 STPA，一些作者得出结论认为，它是一种有效的危险分析方法，尤其适用于评估系统层面组件之间的交互 [37], [38], [52], [54], [57]–[59]。由于面向SAS的STPA扩展了 STPA，因此有理由认为其有效性至少不低于原始 STPA方法。然而，这取决于一个假设，即扩展方法没有以损害有效性的方式偏离原始 STPA；换言之，我们仍然是在进行 *STPA* 吗？我们从两个方面回答这一问题。首先，扩展方法包含原始 STPA 的所有步骤。尽管该扩展增加了一个步骤（适应分类）和参考输入（控制结构、分析义务），但并未删除或修改原有的分析步骤。此外，将扩展方法应用于示例时，忠实地执行了原始 STPA 的步骤，并按预期生成了不安全控制行为和损失场景。其次，扩展方法并不妨碍分析人员按照 STPA 原本的意图应用该方法。第 III-B 节中推导出的 STAMP参考模型采用了 Leveson 的建模构件（过程、控制器、执行器、传感器），并不排斥对现实系统中出现的复杂控制结构进行建模。第III-C 节中的义务描述了针对自适应系统必须至少考虑的因素，但这并不妨碍考虑系统的其他方面。

关于有效性，仍有一个问题：面向*SAS*的*STPA*是否比原始*STPA*方法更适合用于分析*SAS*？回答这个问题超出了本文的范围。一种做法是开展“对照试验”。对照组采用原始STPA，干预组则采用本文所述的方法。可以根据不安全控制行为和损失场景的数量、控制结构的复杂程度以及分析工作量来比较结果。

V. 相关工作

首先，本节回顾现有的自适应系统危险分析方法。随后，将这些现有方法与本文介绍的面向SAS的STPA方法进行比较。

一些作者介绍了仅适用于设计时的自适应系统危险分析方法。Witte 等人结合故障树、攻击树、数据流模型和部署模型，在设计时分析自适应系统被管理系统中的安全与信息安全事件 [28]。他们的故障树和攻击树由专家构建，或根据漏洞数据库生成。他们使用图变换规则描述自适应调整对模型的影响。通过对这些规则进行模型检测，可以检查被管理系统是否会因适应而进入潜在的危险状态。Priesterjahn 等人在设计时使用定时故障传播图（TFPG），评估自愈系统能否在故障导致危害发生之前作出响应 [26]。首先，他们使用UML部署图对被管理系统建模；随后，从人工创建的故障树中推导出最小割集；再为每个割集生成一个TFPG，并分析其时序属性。Alder 等人使用组件故障树、马尔可夫链和MARS适应模型，在设计时计算被管理系统处于危险配置的概率 [25]。最后，Wotawa讨论了使用故障模式与影响分析（FMEA）进行自修复系统设计时分析的方法 [27]。与面向SAS的STPA一样，上述方法仅适用于设计时。

其他作者开发了兼具设计时和运行时阶段的“混合”方法。Aslanesfat 等人在设计时使用故障树和马尔可夫过程模型，并在运行时结合传感器数据，以估算被管理系统发生故障的概率 [31]。Brunner 等人研究了可重构（“即插即飞”）航空电子系统，并提出了涵盖设计时和运行时的工程流程 [30]。在设计时，他们使用故障树分析或故障模式与影响分析来识别每个组件的危险原因，随后将这些原因纳入基于当前系统配置的运行时分析。Dobaj 等人创建了 INSpIRA，这是一种用于分析安全和安保属性的方法 [29], [60]。INSpIRA 使用设计时故障模式与影响分析来发现因果关系，并将其提供给运行时“执行引擎”；该引擎对被管理系统执行测试，以确定是否发生了失效模式。Bhardwaj 和 Liggesmeyer 使用设计时 HAZOP 分析组件“偏差”，并将分析结果存储在系统知识库中，以便在运行时评估与被管理系统新架构配置相关的风险 [61]。Schneider 和 Trapp 提出了条件安全证书（ConSerts），这些证书可以在运行时动态组合成 ConSert 树（类似故障树），以确定被管理系统当前选用的组件能否共同实现整体系统完整性目标 [17], [62]。ENTRUST 方法使用被管理系统和管理系统在设计时及运行时的系统模型，通过模型检验和仿真来验证自适应调整 [18]。在设计时，对管理系统的模型进行验证；在运行时，则验证被管理系统的模型（并考虑当前配置）。验证生成的证据会纳入该系统的动态保证论证。尽管从严格意义上讲，ENTRUST 并非危险分析方法（因为它不会生成危险原因清单），但它与现有方法有许多相似之处，且旨在实现相同的总体目标：保障自适应系统的安全。

所有危险分析方法都包括设计时分析，该分析采用FMEA、FTA或HAZOP等成熟方法。这些方法采用了有限的事故因果关系概念，侧重于组件失效 [32]。相比之下，面向SAS的STPA使用STAMP的

整体事故因果模型，该模型考虑了非故障类危险原因，例如组件交互 [14]。

此外，现有的危险分析方法（ENTRUST 除外）侧重于分析由被管理系统引起的危害原因，并假设管理系统会忠实地执行自适应调整。此外，这些方法在识别危害原因时未考虑 SAS 特定方面。面向SAS的STPA 通过将管理系统及其执行的适应纳入分析范围，克服了这一局限；具体做法是采用适应分类、参考模型和分析义务。

VI. 讨论

本文介绍了对面向自适应系统的STPA 的扩展。该扩展包含一个参考模型和一套推导流程，可用于得出自适应系统的控制结构，并为分析人员提供一组需要考虑的分析义务。该扩展克服了现有危险分析方法的局限，这些局限源于其对事故因果关系的建模方式以及对被管理系统的关注。下文将讨论该方法的工业意义、局限性和效度威胁。

A. 工业意义

除理论贡献外，面向自适应系统的STPA 还具有实际意义。以下论述基于我们在医疗、铁路、汽车、能源和机器人学领域应用传统系统的工业经验。

首先，安全关键系统的开发人员必须开展危险分析，以符合技术标准。必须“按章办事”地应用方法来管理风险。尽管系统理论过程分析（STPA）被公认为一种有效方法，但目前没有指导从业者如何将其应用于自适应系统的指南，也就是说，还没有一本“手册”。本文的贡献可为从业者提供指导。根据我们的经验，提供此类资源可以降低成本并提高危险分析的质量。

其次，完整性是进行危险分析时常见的一个问题，即我们是否考虑周全了？人们常使用引导短语来启发对危险原因的思考（例如，“卡在打开状态”这一短语常用于阀门）。STPA 在识别不安全控制行为和损失场景时会使用引导短语，例如“过迟”和“缺少控制动作” [14]。然而，这些短语并非自适应系统专用。我们的分析义务提供了针对自适应系统的考量，有助于提高对分析结果的信心。

最后，安全需求是工业标准规定的工程过程的基石 [20]、[22]、[63]。为实现合规性，必须针对受管理系统和管理系统规定并验证需求。通过将系统理论过程分析应用于自适应系统，实践人员可以导出这些需求。

B. 局限性与有效性威胁

所提出的方法及其评估存在局限性和有效性威胁，以下将对此进行讨论。

1) 设计时方法： 目前，面向SAS的STPA是一种设计时方法，尚未用于运行时。尽管存在这一局限，我们仍想指出设计时分析的重要性 [64]。首先，从实践角度来看，对于工业利益相关者（监管机构、公众等）而言，仅凭运行时危险分析提供保障，便部署安全关键型自适应系统，可能颇具挑战性，因为现行标准要求进行设计时分析。其次，设计时分析可为后续工程活动提供参考，例如，为关键组件投入更多精力。第三，现有的混合方法依赖设计时分析来建立运行时使用的基线，因此需要设计时方法。第四，从研究角度来看，新的设计时方法可能会为运行时危险分析带来新的见解。最后，并非所有自适应系统都需要运行时分析。如果适应措施定义明确，那么考虑受管理系统和管理系统的设计时方法（如面向自适应系统的STPA）或许就足够了。

2) MAPE-K 架构： 所提出的方法依赖于采用MAPE-K参考模型的管理系统。尽管这是一个广泛使用的模型，但一些自适应系统可能采用其他方法。所提出的方法可能不适用于非 MAPE-K 设计。

3) 可重复性： 尽管我们认为，该方法的某些方面（例如参考模型和分析义务）提高了可重复性，但这一点尚未经过正式评估。不同分析人员得出大相径庭结果的可能性不能排除。

4) 有效性： 所提出方法的有效性尚未经过正式评估。这一对系统理论过程分析的扩展可能并未改善结果（与单独使用系统理论过程分析相比）。尽管基于系统理论过程分析相关研究可以提出一个案例，但由于我们尚未与其他面向自适应系统的方法进行正式比较，因此无法断定该方法优于其他已发表的设计时分析方法。

VII. 结论

本文提出了一种设计时方法扩展，使系统理论过程分析适用于采用 MAPE-K 参考模型的自适应系统。将该方法应用于三个示例所得的结果表明，系统理论过程分析适用于这一类自适应系统。此外，初步评估表明，所提出的方法具有可重复性和有效性，但仍需进一步研究加以确认。

未来工作可聚焦于以下方面：1) 使用本文所述框架开展运行时危险分析；2) 扩展参考模型，纳入 MAPE-K 以外的其他架构；3) 对该方法的可重复性和有效性开展更严格的研究。

参考文献

[1] D. Weyns，《自适应系统导论：当代软件工程视角》。John Wiley & Sons，2020。[2] C. Barna、H. Ghanbari、M. Litoiu 和 M. Shtern，“Hogna：云环境中自适应应用的平台”，载于*2015* 年 *IEEE/ACM* 第*10*届自适应与自管理系统软件工程国际研讨会，2015 年，第83–87页。

[3] S.-W. Cheng、D. Garlan 和 B. Schmerl，“评估 Rainbow 自适应系统的有效性”，载于 *2009* 年 *ICSE* 自适应与自管理系统软件工程研讨会，2009 年，第132–141页。 [4] D. Weyns 和 R. Calinescu，“远程协助：自适应服务型系统示例”，载于 *2015* 年 *IEEE/ACM* 第*10*届自适应与自管理系统软件工程国际研讨会，2015 年，第88–92页。 [5] M. Li 和 Y. Liu，“ 利用无线传感器网络进行地下煤矿监测 ”，*ACM* 传感器网络汇刊，第5卷，第2期，第10:1–10:29页，2009 年。 [6] E. B. Gil、R. Caldas、A. Rodrigues、G. L. G. da Silva、G. N. Rodrigues 和 P. Pelliccione，“人体传感器网络：医疗保健领域中的自适应系统示例”，载于*2021* 年自适应与自管理系统软件工程国际研讨会（*SEAMS*），2021 年，第224–230页。 [7] M. U. Iftikhar、G. S. Ramachandran、P. Bollans´ee、D. Weyns 和 D. Hughes，“DeltaIoT：物联网自适应系统示例”，载于*2017* 年 *IEEE/ACM* 第*12*届自适应与自管理系统软件工程国际研讨会（*SEAMS*），2017 年，第76–82页。 [8] H. Muccini、M. Sharaf 和 D. Weyns，“信息物理系统的自适应：系统性文献综述”，载于 *2016* 年 *IEEE/ACM* 第*11*届自适应与自管理系统软件工程国际研讨会（*SEAMS*），2016 年，第75–81页。 [9] G. Moreno、C. Kinneer、A. Pandey 和 D. Garlan，“DARTSim：用于评估和比较智能信息物理系统自适应方法的示例”，载于*2019* 年 *IEEE/ACM* 第*14*届自适应与自管理系统软件工程国际研讨会（*SEAMS*），2019 年，第181–187页。 [10] S. Gerasimou、R. Calinescu、S. Shevtsov 和 D. Weyns，“UNDERSEA：无人水下航行器自适应系统工程示例”，载于*2017* 年 *IEEE/ACM* 第*12*届自适应与自管理系统软件工程国际研讨会（*SEAMS*），2017 年，第83–89页。 [11] G. Weiss、P. Schleiss、D. Schneider 和 M. Trapp，“在安全关键环境中集成不可靠的自适应系统”，载于第*13*届自适应与自管理系统软件工程国际会议论文集，2018 年，第26–32页。 [12] E. M. Fredericks、B. DeVries 和 B. H. C. Cheng，“面对不确定性时自适应系统测试用例的运行时适应”，载于第*9*届面向自适应与自管理系统的软件工程国际研讨会，丛书 SEAMS 2014。美国纽约：计算机协会，2014 年，第17–26页。 [13] M. U. Iftikhar 和 D. Weyns，“ActivFORMS：用于自适应的主动形式化模型”，载于第*9*届面向自适应与自管理系统的软件工程国际研讨会，丛书 SEAMS 2014。美国纽约：计算机协会，2014 年6月，第125–134页。 [14] N. G. Leveson，《构建更安全的世界：应用于安全的系统思维》。马萨诸塞州剑桥：麻省理工学院出版社，2012 年1月。 [15] S. Diemert 和 J. Weber，“自适应系统中的安全关键自适应”，载于*2022* 年 *IEEE* 国际软件可靠性工程研讨会（*ISSREW*），2022 年10月，第371–380页。 [16] B. Schmerl、J. Andersson、T. Vogel、M. B. Cohen、C. M. F. Rubira、Y. Brun、A. Gorla、F. Zambonelli 和 L. Baresi，“自适应系统保证的组合与分解挑战”，载于《自适应系统软件工程 *III*：保证》，丛书《计算机科学讲义》，R. de Lemos、D. Garlan、C. Ghezzi 和 H. Giese 编。Cham：施普林格国际出版公司，2017 年，第64–89页。[17] D. Schneider 和 M. Trapp，“开放式自适应系统的条件安全认证”，*ACM* 自主与自适应系统汇刊，第8卷，第2期，第8:1–8:20页，2013 年7月。 [18] R. Calinescu、D. Weyns、S. Gerasimou、M. U. Iftikhar、I. Habli 和 T. Kelly，“ENTRUST：通过动态保证论证构建可信自适应软件”，载于第*40*届软件工程国际会议论文集，丛书 ICSE ’18。美国纽约：计算机协会，2018 年，第495页，会议地点：瑞典哥德堡。 [19] R. De Lemos、D. Garlan、C. Ghezzi 和 H. Giese 编，《自适应系统软件工程 *III*：保证》。德国达格施图尔城堡：Springer 柏林海德堡，2017 年，第 III 卷《保证》。

[20] “ISO 26262 - 道路车辆——功能安全”，国际标准化组织，瑞士，标准，2018年。[21] “EN50126 - 铁路应用——可靠性、可用性、可维护性与安全性（RAMS）的规范与论证——第1部分：通用RAMS过程”，欧洲电工标准化委员会，标准，2017年。[22] “IEC61508 - 电气／电子／可编程电子安全相关系统的功能安全”，国际电工委员会，日内瓦，技术报告IEC61508，2010年。[23] “安全关键系统软件开发专业实践指南”，不列颠哥伦比亚省工程师与地球科学家协会，加拿大不列颠哥伦比亚省本拿比，指南，2020年。[24] C. Ericson，《系统安全危险分析技术》，第2版。新泽西州霍博肯：威利出版社，2015年7月。[25] R. Adler、M. Forster和M. Trapp，“确定安全关键自适应系统的配置概率”，载于第21届高级信息网络与应用国际会议研讨会（AINAW’07），第2卷，2007年，第548–555页。[26] C. Priesterjahn、D. Steenken和M. Tichy，“自愈系统的基于组件的定时危险分析”，载于第8届自适应系统保障研讨会论文集，丛书ASAS ’11。美国纽约州纽约市：计算机协会，2011年，第34–43页。[27] F. Wotawa，“自适应与自主系统的第一性原理推理”，载于《动态系统预测性维护：高级方法、决策支持工具与现实世界应用》，E. Lughofer和M. Sayed-Mouchaweh编。Cham：施普林格国际出版公司，2019年，第427–460页。[28] T. Witte、R. Groner、A. Raschke、M. Tichy、I. Pekaric和M. Felderer，“面向跨自适应步骤的模型协同演化：结合安全与信息安全分析”，载于2022年自适应与自管理系统软件工程国际研讨会（SEAMS），2022年，第106–112页。[29] J. Dobaj，“INSpIRA：将信息安全整合到风险评估中：博士项目论文”，载于第13届自适应与自管理系统软件工程国际会议论文集，丛书SEAMS ’18。美国纽约州纽约市：计算机协会，2018年，第183–187页。[30] M. Brunner、D. N. Darwesh和B. Annighoefer，“面向自适应安全关键即插即飞航空电子系统的安全流程”，载于2021年IEEE/AIAA第40届数字航空电子系统会议（DASC），2021年，第1–10页。[31] K. Aslansefat、P. Nikolaou、M. Walker、M. N. Akram、I. Sorokos、J. Reich、P. Kolios、M. K. Michael、T. Theocharides、G. Ellinas、D. Schneider和Y. Papadopoulos，“SafeDrones：使用可执行数字可信身份对无人机进行实时可靠性评估”，载于《基于模型的安全与评估》，丛书《计算机科学讲义》，C. Seguin、M. Zeller和T. Prosvirnova编。Cham：施普林格国际出版公司，2022年，第252–266页。[32] N. Leveson，“用于构建更安全系统的新事故模型”，《安全科学》，第42卷，第4期，第237–270页，2004年。[33] S. Apel、J. M. Atlee、L. Baresi和P. Zave，“特征交互：下一代”，《达格施图尔报告》，第4卷，第24期，2014年，第7–11页。出版地：德国达格施图尔；出版者：Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik。 [在线]。可访问：http://drops.dagstuhl.de/opus/volltexte/2014 4783 / M. Trapp和D. Schneider，“开放式自适应系统的安全保障——综述”，载于《Models@run.time：基础、应用与路线图》，丛书《计算机科学讲义》，N. Bencomo、R. France、B. H. C. Cheng和U. Aßmann编。Cham：施普林格国际出版公司，2014年，第279–318页。[35] J. Kephart和D. Chess，“自治计算的愿景”，《Computer》，第36卷，第1期，第41–50页，2003年，会议名称：Computer。[36] D. Weyns、B. Schmerl、V. Grassi、S. Malek、R. Mirandola、C. Prehofer、J. Wuttke、J. Andersson、H. Giese和K. M. G¨oschka，“自适应系统去中心化控制的模式”，载于《自适应系统软件工程II：国际研讨会，德国达格施图尔城堡，2010年10月24日至29日，经修订的精选与特邀论文》，丛书《计算机科学讲义》，R. de Lemos、H. Giese、H. A. M¨uller和M. Shaw编。柏林、海德堡：施普林格，2013年，第76–107页。[37] X.-Y. Zhou、Z.-J. Liu、F.-W. Wang、Z.-L. Wu和R.-D. Cui，“自主船舶危险分析方法的适用性评估”，《海洋工程》，第214卷，文章编号107773，2020年。[34]

[38] Blake R. Abrecht，“系统理论过程分析在海上供应船动态定位系统中的应用”，硕士学位论文，麻省理工学院，美国马萨诸塞州剑桥，2014年。[39] W. Young和N. Leveson，“面向安全与信息安全的系统思维”，载于第*29*届年度计算机安全应用会议论文集，丛书ACSAC '13。美国纽约州纽约市：ACM，2013年，第1–8页。[40] W. Young和N. G. Leveson，“基于系统理论的安全与信息安全一体化方法”，《*ACM*通信》，第57卷，第2期，第31–35页，2014年。[41] J. H. Weber-Jahnke和F. Mason-Blakley，“电子病历（EMR）系统的安全性：EMR安全意味着什么？我们如何构建更安全的系统？”《*SIGHIT Rec.*》，第1卷，第2期，第13–22页，2011年。[42] J. H. Weber和O. Costa，“为电子健康互操作性符合性配置文件的分析调整系统理论危险分析方法”，《*AMIA*转化科学峰会论文集》，第2020卷，第693–702页，2020年。[43] “IEEE 1228 - IEEE软件安全计划标准”，IEEE标准协会，标准，1994年。[44] S. Jahan、A. Marshall和R. Gamble，“维持安全性论证案例的自适应策略”，载于第*12*届*IEEE*自适应与自组织系统国际会议，*SASO 2018*，*2018*年*9*月*3*日至*2018*年*9*月*7*日，丛书《自适应与自组织系统国际会议》，SASO，第2018年9月卷。意大利特伦托：IEEE计算机学会，2018年，第180–185页。[45] E. Asaadi、E. Denney、J. Menzies、G. J. Pai和D. Petroff，“动态保证案例：通往可信自主性的路径”，《*Computer*》，第53卷，第12期，第35–46页，2020年12月。[46] N. G. Leveson和J. P. Thomas，“STPA手册”，技术报告，2018年。[47] John P. Thomas，“扩展并自动化用于需求生成与分析的系统理论危险分析”，学位论文，麻省理工学院，美国马萨诸塞州剑桥，2013年。[48] Margaret Stringfellow，“面向人类与组织因素的事故分析和危险分析”，学位论文，麻省理工学院，美国马萨诸塞州剑桥，2010年。[49] Megan E. France，“面向人类的工程：STPA的新扩展”，硕士学位论文，麻省理工学院，美国马萨诸塞州剑桥，2017年。[50] D. Weyns、M. U. Iftikhar、D. Hughes和N. Matthys，“应用基于架构的适应来自动化物联网管理”，载于软件架构，丛书《计算机科学讲义》，C. E. Cuesta、D. Garlan和J. P´erez编。Cham：施普林格国际出版公司，2018年，第49–67页。[51] S. Gerasimou、R. Calinescu和A. Banks，“利用缓存、前瞻和近最优重新配置进行高效的运行时定量验证”，载于第*9*届自适应与自管理系统软件工程国际研讨会论文集，丛书SEAMS 2014。美国纽约州纽约市：计算机协会，2014年，第115–124页。[52] T. Ishimatsu、N. G. Leveson、J. Thomas、M. Katahira、Y. Miyamoto和H. Nakao，“使用STPA进行建模与危险分析”，载于第*4*届*IAASS*会议论文集，让安全至关重要。阿尔伯塔省亨茨维尔：国际空间安全促进协会（IAASS），2010年。[53] Elizabeth W. Baker，“应用STAMP过程进行医院用药管理中的安全分析”，硕士学位论文，麻省理工学院，美国马萨诸塞州剑桥，2022年。[54] Rodrigo S. Mart´ınez，“面向汽车应用的电动助力转向系统理论过程分析”，硕士学位论文，麻省理工学院，2015年。[55] Sarah A. Folse，“爱德华兹空军基地小型无人航空系统使用的系统理论过程分析”，硕士学位论文，麻省理工学院，美国加利福尼亚州爱德华兹空军基地，2017年。[56] D. G. Feitelson，“从可重复性到可再现性与佐证”，《*ACM SIGOPS*操作系统评论》，第49卷，第1期，第3–11页，2015年。[57] L. Sun、Y.-F. Li和E. Zio，“HAZOP、故障模式与影响分析、FRAM和STPA方法在自动紧急制动系统危险分析中的比较”，《*ASCE-ASME*工程系统风险与不确定性杂志*B*辑：机械工程》，第8卷，第3期，2021年。[58] S. M. Sulaman、A. Beer、M. Felderer和M. H¨ost，“故障模式与影响分析和STPA安全分析方法的比较——案例研究”，《软件质量杂志》，第27卷，第1期，第349–387页，2019年。

[59] Pawlicki Todd、Samost Aubrey、Brown Derek W.、Manger Ryan P.、Kim Gwe-Ya 和 Leveson Nancy G.，《将基于系统与控制理论的危险分析应用于放射肿瘤学》，医学物理学，第 43 卷，第 3 期，第 1514–1530 页，2016 年。[60] J. Dobaj 、 J. Iber 、 M. Krisper 和 C. Kreiner，《迈向可执行的可信性属性》，载于 系统、软件与服务过程改进，《计算机与信息科学通信》系列，X. Larrucea、I. Santamaria、R. V. O'Connor 和 R. Messnarz 编，Cham：施普林格国际出版公司，2018 年，第 341–353 页。[61] N. Bhardwaj 和 P. Liggesmeyer，《开放式自适应系统中安全重新配置的运行时风险评估概念》，载于 计算机安全性、可靠性与安保，《计算机科学讲义》系列，S. Tonetta、E. Schoitsch 和 F. Bitsch 编，Cham：施普林格国际出版公司，2017 年，第 309–316 页。[62] D. Schneider，《开放式自适应系统的条件安全认证》，学位论文，凯泽斯劳滕工业大学、弗劳恩霍夫IESE研究所，2014 年。[63] 《RTCA DO-178C——机载系统与设备认证中的软件注意事项》，航空无线电技术委员会，美国华盛顿特区，标准 DO-178C，2011 年。[64] S. Morisaki 和 N. Kasai，《自动化自适应软件系统中关键适应的检测方法》，载于 *2018 IEEE* 软件质量、可靠性与信息安全国际会议伴会（*QRS-C*），2018 年，第 526–530 页。