之前 自主性接管控制：机器人软件测试

NILS CHUR，德国波鸿鲁尔大学THIAGO SANTOS DE MOURA，德国波鸿鲁尔大学
ARGENTINA ORTEGA，德国不来梅大学和德国波鸿鲁尔大学SVEN PELDSZUS，德国波鸿鲁尔大学和丹麦哥本哈根信息技术大学THORSTEN BERGER，德国波鸿鲁尔大学和瑞典查尔姆斯|哥德堡大学NICO HOCHGESCHWENDER，德国不来梅大学YANNIC NOLLER，德国波鸿鲁尔大学

arXiv:2602.02293v1 [cs.SE] 2 Feb 2026

机器人系统是复杂的安全关键软件系统，因此需要进行全面测试。遗憾的是，与传统软件相比，机器人软件本质上更难测试，主要是因为软件需要与硬件密切交互、应对运行环境中的不确定性、处理扰动，并以高度自主的方式行动。然而，鉴于机器人运行的空间范围广阔，在设计测试时预测可能出现的失效颇具挑战。本文通过考察机器人测试论文并将其与软件测试理论联系起来，开展了一项映射研究。我们考察了247篇机器人测试论文，并将其映射到软件测试领域；通过一个示例阐述机器人软件测试的最新进展，并讨论当前的挑战。本文旨在为机器人学和软件工程领域的研究群体介绍软件测试挑战。最后，我们指出了尚待解决的问题和经验教训。

CCS 概念：• **计算机系统组织** → **机器人学**；• **软件及其工程** →**软件测试与调试**。

附加关键词和短语：服务机器人学，软件测试

1 引言

机器人系统日益普及，应用于医疗保健、农业和交通等各种领域，因此可靠性和安全性至关重要。这些系统结合硬件和软件组件，能够在动态且安全关键的环境中自主运行 [72]。因此，部署前必须进行全面测试，以发现失效——“系统无法在规定的性能要求范围内执行其所需功能” [97]——同时还要评估性能和标准符合性，并确保其可信性和可靠性。测试不足可能导致人员受伤、基础设施受损以及利润损失。

测试机器人系统面临着独特的挑战，因为软件必须应对传感器噪声、执行器变异以及其他会引入非确定性的外部影响，这些因素可能导致行为错误或意外失效 [1]。除了机器人学特有的挑战 [39, 73, 74, 191], 之外，机器人软件还面临着与传统软件系统相同的挑战，需要各种组件在不确定且不断变化的条件下无缝协同工作。因此，要在各种环境和运行条件下验证机器人系统的行为仍然颇具挑战。机器人测试被认为较为零散且大多缺乏针对性，往往临时开展 [12], ；而软件工程已制定了全面的软件测试标准 [98, 102–104], ，但这些标准如何应用于机器人学仍不明确。据我们所知，目前尚无

作者地址：Nils Chur，nils.chur@rub.de，波鸿鲁尔大学，德国波鸿；Thiago Santos deMoura，thiago.santosdemoura@rub.de，波鸿鲁尔大学，德国波鸿；Argentina Ortega，argentina.ortega@uni- bremen.de，不来梅大学，德国不来梅及波鸿鲁尔大学，德国波鸿；Sven Peldszus，sven.peldszus@rub.de，波鸿鲁尔大学，德国波鸿及哥本哈根信息技术大学，丹麦哥本哈根；Thorsten Berger，thorsten.berger@rub.de，波鸿鲁尔大学，德国波鸿及查尔姆斯|哥德堡大学，瑞典哥德堡；Nico Hochgeschwender，nico.hochgeschwender@uni-bremen.de，不来梅大学，德国不来梅；YannicNoller，yannic.noller@rub.de，波鸿鲁尔大学，德国波鸿。

系统综述将机器人测试与软件测试标准联系起来。当前文献聚焦于特定应用、案例研究或单一测试方法，却缺乏对测试流程或适当技术的全面综述。尽管近期研究考察了机器人领域的软件工程实践 [37, 38, 72] 以便更好地理解机器人测试面临的独特挑战 [1, 2, 12, 235], 但很少有研究对机器人测试作简明介绍。

我们通过一项关于机器人测试的系统性文献综述来填补这一空白，分析软件工程中的哪些测试技术与标准已被应用以及如何应用。我们的目标是将既有的软件测试理论引入机器人学界，同时向软件工程界呈现机器人测试的独特特征，从而识别差距与挑战。

2 背景与动机

机器人系统由紧密耦合的软件与硬件组件构成，例如规划与控制模块、传感器和执行器。其跨学科性质涵盖计算机科学、控制理论以及机械与电气工程，使测试更为复杂；测试必须考虑异构组件及其错综复杂的交互，从而产生大量潜在失效点。这一挑战因机器人开发中常常缺乏规范的软件工程实践而进一步加剧，限制了严格测试的应用 [72]。

测试不足会增加失效的可能性，即软件系统无法执行其预期功能，例如产生错误输出。失效源于底层故障，例如错误或缺失的代码 [97], , 并通过测试加以识别。在机器人学中，失效往往源于软件组件之间的交互、单个组件故障或不可预测的环境。中间件的使用虽有助于组件集成，却引入了额外的复杂性，使得根因分析对于定位失效的确切来源至关重要。与可能导致崩溃或错误输出的传统软件失效不同，机器人软件失效可能造成人身伤害或财产损失。在医疗保健或自动驾驶等领域与人类协同运行，使这些系统成为安全关键系统 [33]，因而需要严格的安全验证，且往往须符合严格的行业标准 [26]。

2.1 测试

软件测试可分为静态测试和动态测试：前者在不执行软件的情况下对其进行检查，后者则通过运行软件来检测失效。本文聚焦于动态测试，Lochau 等人 [142]将其定义为：“在从通常无限的输入/执行域中适当选取的有限测试用例集上，对照预期行为，对程序行为进行动态验证/验证。”在此语境下，测试用例是离散、可重复的单元，能够对软件行为进行系统性评估 [103]。每个测试用例包含四个要素：

(I) 前置条件：测试执行前的状态，包括机器人环境、数据、软件和硬件 [103]。这包括机器人的初始位姿、位姿估计置信度、地图、系统配置和测试环境。我们区分环境表示（即对机器人周围环境进行建模的输入）与测试环境；后者包括执行一组测试所需的全部资源，例如模拟器、硬件设置和中间件配置。

（*II*）测试输入：执行测试所需的数据，例如应用于系统的命令或激励，范围从简单的函数参数到完整的仿真环境。

（*III*）预期结果：测试执行期间预期的系统行为或状态，用于判断是否已达到测试目标。

（*IV*）后置条件：测试完成后系统和环境的状态，包括“成功标准、需要检查的失效等”。[103], 例如确认信号、显示正确行为的日志，或表明测试失效的错误消息。

表1. 我们运行测试的需求 ple

编号 功能	需求
1 接收来自应用程序/厨房工作人员的食品订单	机器人应将新的“取餐和配送”任务添加到其任务队列中。
2 通过网络接收前往餐桌的请求。机器人应将“接收订单”任务以高优先级添加到任务队列中。	
3 识别有顾客准备点餐的桌子	机器人应识别餐桌旁顾客的手势。
4 通过机器人上的触摸屏接收订单	机器人应告知顾客已收到订单。
5 通过语音接收订单	机器人应听取订单并创建机器可读的食品订单任务。
6 从厨房取餐	机器人应等待确认食品订单已正确装载。
7 将订单送至餐桌	机器人应检测到食品订单已卸载。 机器人应请求确认食品订单已正确送达。
8 拾取餐具和盘子	机器人应自主收集留在空餐桌上的所有盘子和餐具。 机器人应将脏盘子和餐具放入水槽。
9 清洁表面	机器人应手持海绵，沿网格路径移动其机械臂。
10 导航	机器人应能够自主到达部分预定义位置（例如顾客餐桌、厨房取餐和送餐地点）。 机器人不得撞到人。 机器人的速度不得超过 0.8 m/s 的最高安全速度。
11 电池充电	在整个运行过程中，机器人应将“电池充电”任务添加到其任务队列，以维持其总电池容量的至少40%。

机器人系统测试仍然充满挑战，因为它们运行在高度多变且连续的环境中，这与部署在更受控环境中的其他信息物理系统不同。这使得系统地推导边界情况 [1] 并明确预期行为 [4]变得复杂。尽管信息物理系统测试已有相关研究 [75], 但针对机器人系统特定特征的实用指导仍然有限，尤其是对于缺乏软件测试背景的从业者，或缺乏机器人技术专业知识的软件工程师而言。本文旨在弥合这一差距，概述与机器人测试相关的关键概念、测试级别、类型和活动，并辅以具体示例。

2.2 案例研究：餐厅机器人

作为示例，我们的运行示例是一台在餐厅工作的服务机器人。它的设计灵感来自 RoboCup@Home 竞赛中的挑战。该竞赛旨在推动个人服务机器人的开发，并在真实的家庭环境中对其性能进行基准测试。其他灵感则来自 Bella 和 Servi 等商业机器人。

餐厅机器人： 机器人通过协助服务员共同接收和递送订单、清理餐桌来提供帮助。该机器人是 PAL Robotics TIAGo（图1），配备差速驱动底盘、用于定位和障碍物检测的激光传感器、带平行夹爪的 7 自由度机械臂、车载计算机、用于感知的 RGB-D 摄像头，以及用于用户交互的扬声器、麦克风阵列和触摸屏。



图 1. PAL Robotics TIAGo 机器人 [147]

任务。我们重点关注三项任务：机器人必须能够在餐厅内自由移动，同时避开与家具和人员发生碰撞。机器人的速度必须保持安全，并在运送食物或液体时做出调整。机器人应监控电池电量并及时充电，以保持可用状态。表1 列出了实现这三项任务所需的（不完整）任务需求清单。

1. 接收顾客订单。 机器人通过应用程序接收顾客订单，顾客也可以通过应用程序或手势召唤机器人。订单可以通过机器人的平板电脑输入，也可以通过语音识别录入。厨房工作人员可以要求机器人取送订单至指定餐桌。

2. 订单的取货与配送。机器人从当前位置导航至预先设定的取货点，由工作人员装载订单。机器人会等待确认订单已成功装载后再继续行驶。

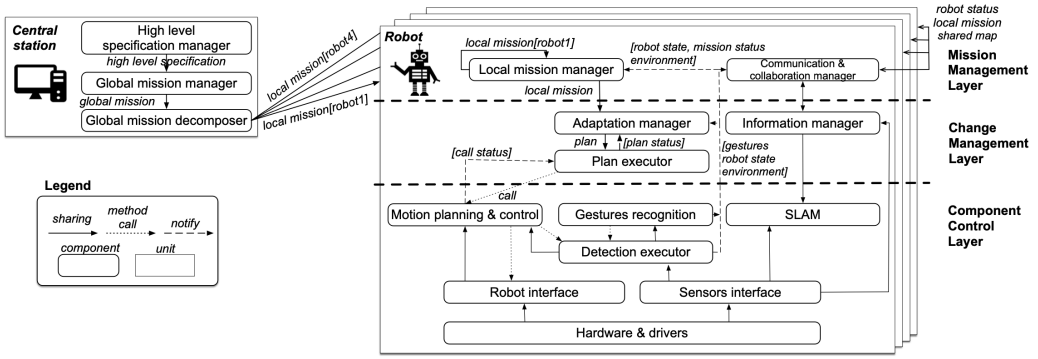


图 2. 自适应去中心化机器人架构（SERA）[71]

确认后，机器人会从厨房导航至指定餐桌，并等待工作人员或顾客卸下订单。

3. 清理桌子。 机器人会收拾剩余的餐具，将它们放到托盘上，再送回厨房。它还可以用海绵清洁桌子，并用抹布擦干。**软件**。我们的机器人使用机器人操作系统（ROS）[202],，这是一种中间件和机器人应用程序框架。ROS 应用程序由所谓的节点（即独立的软件组件）组成，这些节点通过发布-订阅及其他通信机制进行通信。例如，LaserNode 会将传感器读数发布到一个所谓的 ROS 主题，而 LocalizationNode 会订阅该主题，并在收到新数据时进行计算。

在我们的示例机器人中，我们采用自适应去中心化机器人架构（SERA）[71],，这是一种包含任务管理、变更管理和组件控制的三层架构（图 2）。在任务管理层，会处理服务目标（例如将订单送到指定餐桌），并将其分解为可执行的任务，由本地任务管理器负责协调。变更管理层会监控任务执行，并对环境或运行状况的变化作出响应，例如路径受阻或与人交互。适应管理器会评估这些情况，并在需要时触发重新规划或重新调度，而计划执行器则将任务分派给底层控制组件。组件控制层包括导航、运动控制和感知组件，并与硬件和传感器交互。

每个组件都由多个节点构成，这些节点跨层与其他节点交互，形成系统之系统。例如，LocalizationNode 处理来自 LaserNode 的激光扫描数据，也处理里程计数据，维护机器人位姿的概率信念，并持续发布供导航进行避障使用的位姿估计。类似地，运动规划与控制组件同时订阅定位更新、地图数据和障碍物信息，同时向计划执行器组件发布控制命令和状态反馈。节点之间不同的更新频率，以及彼此独立且并发的执行，进一步增加了消息交换的复杂性。因此，单个组件的行为会受到其他组件、当前任务和适应上下文的影响。通过异步通信形成的紧密耦合，加上跨层并行执行以及通常所需的实时能力，说明了机器人软件为何如此复杂。

3 方法论

我们的方法论结合了专家知识与文献综述，以回顾机器人测试的最新进展。具体而言，我们将 ISO 测试标准 [98, 102–104], 所呈现的软件测试最新进展，与机器人测试文献中的测试实践进行了比较。随后，我们基于专家知识对两者的重合部分进行了分析。为此，我们邀请了

多个相关领域的专家共同参与。最终的作者团队由机器人学、一般测试以及软件工程领域的专家组成，其中大多数人精通多个领域。这些专业知识使我们不仅能够全面概述相关内容，还能结合实际经验，将现有文献及作者熟悉的实践置于具体背景中。除此之外，这种互补的专业知识还使我们得以识别机器人软件测试最新进展中的现有空白。

我们利用ISO29119标准族（主要是第1–4部分）中关于测试的知识体系 [98, 102–104]，系统性地识别社区认为相关的测试方面。基于我们在测试方面的专业知识以及补充文献，例如ISTQB认证测试员的文档 [100, 101] 以及学术测试论文或书籍 [21, 88, 142],，所有作者讨论了应以何种抽象层次来考量测试方面，并决定例如合并ISO标准中引入的所有低层覆盖准则。

图3 展示了我们的策略。我们在Scopus中检索聚焦于测试机器人系统的论文，即那些介绍或评估测试工具、提出测试技术，或报告明确涉及验证与确认的案例研究的论文。我们还纳入了讨论与测试相关的挑战、局限性或开放性问题的研究。为涵盖测试与机器人学的所有可能维度，我们使用通用检索字符串*"RobotANDTest"*在论文标题和摘要中识别论文。我们将检索限定于"计算机科学"和"工程"类别，并要求论文具有任何与机器人相关的作者关键词，例如导航、操作或路径规划——这些关键词是通过浏览Scopus中全部161个关键词的完整列表选出并加入我们的检索词的。为确保质量，我们仅考虑发表在期刊或会议论文集中、篇幅至少六页的完整论文。该检索得到21,068篇可能报告机器人测试或面向机器人的测试技术的论文。为识别实际上报告机器人测试的论文，我们采用了两步筛选，首先依据标题，然后依据摘要。我们由两位作者各自独立筛选了60篇论文，其中50%在是否排除或纳入论文上达成一致。分歧

由所有作者共同讨论。由于一致性较低，且标题所含信息有限，我们决定仅排除明显不符合条件的论文。根据标题进行初步筛选后，剩下 705 篇论文。接下来，我们根据摘要进行了第二轮筛选。这一轮中，每篇论文由两位作者进行筛选，160 篇论文的筛选结果中有 71% 达成一致。对于意见不一致的情况，所有作者再次进行讨论，随后各自独立筛选论文；根据摘要筛选后，共纳入 249 篇论文。

我们将测试社区的测试相关概念与已识别的论文进行了映射，从而概览了机器人学界涵盖了哪些测试方面。因此，我们排除了17篇无法访问的论文、12篇篇幅低于我们的篇幅限制（元数据中未注明）的论文，以及28篇尽管摘要表明其相关，但并未描述与测试知识体系相关的机器人测试方面的论文。最终，我们得到198篇关于机器人学测试的论文，其中12篇报告了测试挑战，其余186篇则映射到了测试知识体系。此外，我们还利用 Google Scholar，根据我们的专业知识，对测试知识体系的所有方面进行了机器人学论文的人工检索，

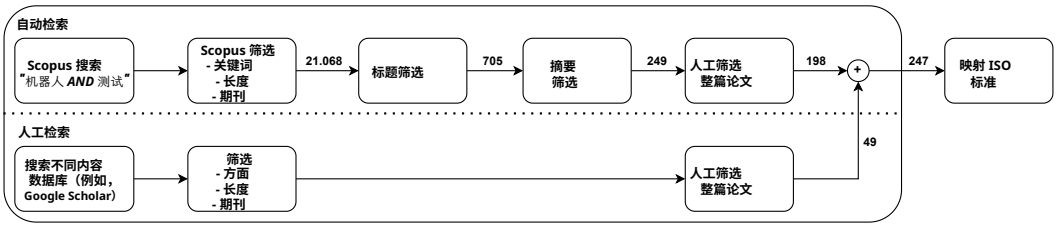


图 3. 搜索策略和筛选流程概览

尤其针对初始检索未涵盖的方面，因此又找到49篇论文。为扩大此次检索的范围，我们有意没有再次使用 Scopus。

基于测试知识体系与机器人学文献之间的映射、论文中的详细信息，以及我们在机器人学领域的经验，我们总结了机器人领域测试的最新进展。由此，我们指出了机器人社区尚未涉及的测试方面、对特定测试方面的理解存在的分歧，以及虽有涉及但未像测试社区所期望的那样系统化的方面。这些观察构成了深入讨论的基础；我们尤其借助作者不同背景所带来的不同视角，提出了改进机器人系统测试的建议。

4 测试环境、级别、类型和活动

测试由一系列共同构成测试过程的活动组成 [24], 这些活动包括决定测试什么、如何测试以及在何种环境中测试。我们遵循软件测试标准提出的典型测试分类 [98, 102–104]。首先，我们在第4.1节介绍测试环境，即执行测试的情境。随后，我们在第4.2节介绍测试级别，这些级别代表与系统开发过程相对应的不同测试阶段。每个测试级别都侧重于系统的不同部分，从单个单元到整个系统。接着，我们在第4.3节介绍测试类型，说明测试的目的，以及测试在各种条件下是否实现了预期功能或质量。最后，我们将在第4.4节讨论测试活动，介绍从规划、实施到执行和评估的系统化测试过程。

我们发现，见图 4，大多数论文关注测试环境（59篇），其中大部分采用基于仿真的测试（43篇），其次是真实世界测试（13篇）。在测试类型方面，大多数论文讨论非功能测试（33篇），重点关注非功能性能测试（15篇）；另有论文讨论功能测试（14篇）。关于测试级别（26篇）的研究较少，这些测试级别分为集成测试（7篇）、验收测试（6篇）、端到端测试（4篇）和单元测试（3篇）；关于测试活动的研究（5篇）也很少。

4.1 测试环境

执行测试的环境，是机器人测试中一个特定的方面，相较于
与传统软件系统类似。它可以是模拟运行环境的基于仿真的环境，也可以是真实环境。测试环境的选择会影响多个方面。虽然基于仿真的环境在成本和速度方面具有优势，但其真实感不及在真实环境中进行测试 [2, 174, 206, 237]。

基于仿真的测试。基于仿真的测试是指在虚拟环境中执行测试，其环境范围从简单的模拟对象到高保真仿真不等。开发的软件会在机器人及其传感器、执行器和动态环境的虚拟模型上运行。由于具有可扩展性、可重复性和成本效益，基于仿真的测试在机器人学中得到广泛应用，尤其适用于无法进行真实世界测试的领域 [2, 138, 189, 237]。

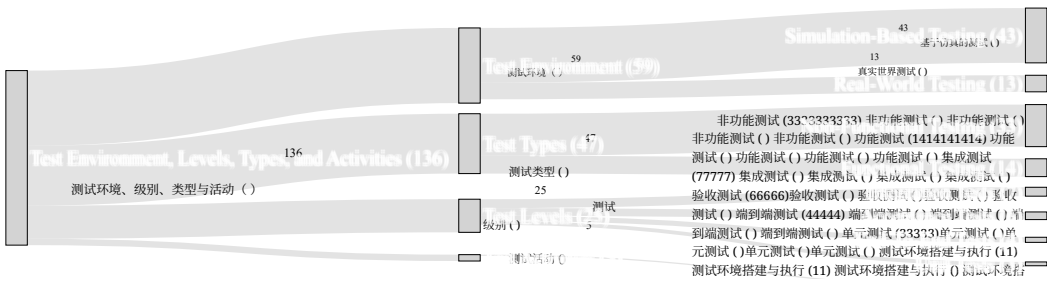


图 4. 报告相应测试方面的机器人学论文数量

这种方法能够在不承担真实世界测试相关风险（例如财产损失）的情况下测试各种场景，并支持自动化评估和回归测试 [10, 109, 220, 246]。

在虚拟环境中进行测试的缺点是逼真度有限，因为它无法完全复现真实世界的复杂性。传统软件测试通常依靠虚拟环境来模拟执行条件，而机器人仿真则难以模拟真实世界物理特性、传感器误差和动态环境条件 [204, 225]。摩擦、材料变形和传感器噪声等因素也会导致仿真性能与真实世界性能之间出现差异，这种差异也称为仿真—现实差距 [5]。

这方面的研究侧重于构建仿真，通过为不同的机器人（例如机械臂、人形机器人 [134, 194, 212, 227, 269], 、地面车辆 [228, 232], 或航空器 [148]）设计高保真仿真来应对仿真—现实差距。由此产生了许多简化仿真设置或机器人集成的框架 [57, 171, 230, 268], ，适用于地面车辆 [14, 222, 236, 252], 和空间机器人 [110]。仿真还可用于评估机器人系统 [164] 或各类组件，涵盖控制 [45, 47, 99, 150, 168, 219], 、协作和编队 [160, 174, 226], ， 以及感知 [83] 和导航 [159, 193]。此外，研究人员还会测试机器人在人机交互 [92, 278], 或多机器人系统 [54, 87, 206]中的安全性。不过，这仅展现了复杂性的一小部分，并提供了一个可靠的基础；全面掌握基于仿真的测试的复杂性超出了本文的讨论范围。**真实世界测试**。通过在真实的机器人硬件上执行软件，并让机器人直接在物理环境中运行，真实世界测试对于验证实际性能 [151, 274] 和行为是否正确 [125, 155, 186, 253]至关重要。它有助于识别仿真中采用的假设、硬件集成相关问题以及环境变化。相关研究致力于为系统开发基准测试，以便更好地比较不同解决方案 [30, 254],, 并评估人机交互中的安全性 [199, 248]。真实世界测试通常用于较高的测试级别，因为这需要完整的系统。实际测试环境不一定是机器人设计所针对的环境，也可以是实验室或受控的真实世界环境。将机器人部署到其设计运行的环境中，是一种特殊的真实世界测试，通常称为基于现场的测试，详见第 5.5 节。

真实世界测试具有最高的保真度和真实性，但成本高、可重复性较低且耗时 [229]。结合使用基于仿真的测试和真实世界测试，可以兼顾保真度与成本 [281]。在开发期间进行基于仿真的测试，在验证阶段进行真实世界测试 [1, 2, 149]。在下文中，我们将重点讨论不同的测试方面；测试环境在其中起次要作用，因此不再专门讨论。

4.2 测试级别

测试级别根据被测系统的范围和粒度对测试活动进行分类，明确测试的实施位置，以及目标如何与该范围相匹配。

测试级别最好用V模型来说明。如图5所示，它是一种强调测试、并广泛应用于安全关键领域（包括机器人学）的软件开发过程模型 [27, 140]。它突出了开发活动（V的左手侧）与测试活动（V的右手侧）之间的联系。每一项开发活动都有直接对应的测试活动，用于根据开发规格说明评估系统 [210]。V模型只是一种过程模型；实践中的实际开发进程会结合不同的模型，尤其是与迭代增量过程模型相结合。许多敏捷过程同样是迭代增量的，并强调测试，尤其是自动化测试。这些进程通常以模型驱动设计和基于组件的设计为指导，二者侧重于在整个系统生命周期中进行复用、形式化和验证 [223, 224]。在此，更高层次的描述，即系统如何工作的模型

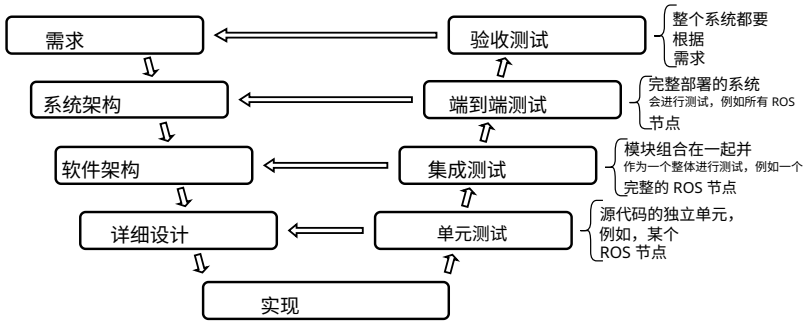


图5. V模型中的测试级别以及机器人系统相关部分的示例

（其结构与行为）会首先被创建。随后，这些模型驱动或自动化实现，有助于管理复杂性并简化更新，因为代码可以重新生成，而无需重写。

Lim等人 [140]提出了一种分层测试模型，即针对与硬件模块集成的机器人软件组件所设立的一组有组织的测试级别（单元、集成和系统）。该模型基于V模型，并采用黑盒测试来展示其在嵌入式机器人测试平台上的效率。Nagrath 和 Schlegel [173] 介绍了 SmartTS，这是一种基于组件且由模型驱动的方法，可自动执行所有级别的测试，并生成可验证的测试记录（即通过根据模型导出的契约检查组件行为而获得的结果），从而提高可信性。

单元测试。在 V 模型的最低层级，单元测试用于验证小型单元（例如代码函数）是否正确实现。它逐一检查每个单元，确保其行为符合预期。单元测试对应 V 模型中的详细设计 活动，在这一阶段会对每个单元进行规范定义，并依据规范进行验证。单元测试可以在开发早期开展。原则上，可以为每个代码函数编写测试；尽早测试有助于及早发现并修复故障，从而降低工程成本 [158]。

在基于ROS的系统中，单元测试检查其中的小片段（例如单个函数），而非测试整个节点本身 [28]。这种粒度符合通用的软件测试实践，即单元测试针对彼此隔离的功能。为此，ROS 文档建议将核心业务逻辑与 ROS 专属代码（如回调或通信句柄）解耦，以便使用标准框架进行测试，例如用于 C 的`gtest` ++或用于 Python 的`unittest` 。此外，文档还建议在实施修复之前编写测试以复现错误，并使用模拟器或模拟对象替代硬件，以确保结果具有确定性且可重复 [207, 208]。

为了促进机器人学中的自动化测试，Bihlmaier 和 Wörn [28] 提出了机器人单元测试（RUT）方法论。RUT 在 ROS 框架中使用基于传感器和执行器数据的仿真来测试软件组件和机器人行为。这确保了高度自动化、独立性和并行执行。Paikan 等人 [184] 提出了一个通用框架，可独立于编程语言和中间件开发并运行单元测试。测试以插件形式实现，便于复用并分组为测试套件。测试夹具管理器会设置所需的环境，例如模拟器或机器人驱动程序，并监控资源，以确保测试执行的一致性。

集成测试。这类测试用于确定系统中的各个单元能否正确地相互交互，使开发人员能够对它们组合后的效果建立信心，同时验证其实现是否满足需求。单元测试针对方法或函数，而集成测试则关注粒度更粗的组件，例如整个 ROS 节点。

在 ROS 中，集成测试对应 V模型的架构层。在这一层，测试会验证模块（例如方法或函数）能否协同工作，形成一个正常运行的节点，以及多个节点能否正确交互，从而提供实际功能（例如话题、服务、动作）。例如，感知节点可能会发布数据，供规划节点使用。根据

文档说明，开发人员可以使用rostest （ROS 1）或 launch_测试 （ROS 2）在受控场景中运行节点。在这一阶段发现的常见问题包括消息类型不匹配、服务接口错误、竞态条件和资源冲突 [207, 208]。

Brito 等人 [35] 提出了一种基于图的方法，将发布/订阅交互表示为通信图，并结合功能（黑盒）测试和结构（白盒）测试来检测消息级故障，例如丢失、重复和损坏。他们的实验表明，缺陷检测和覆盖率均显著提升。此外，Piel 等人 [192] 介绍了两种针对发布/订阅网络的自动化集成测试方法。一种方法会生成大量事件序列，以发现发生故障的系统状态；另一种则使用预定义的事件序列来验证系统行为。Kang 和 Park [114] 针对复合组件的测试预言机问题开展研究。他们开发了一种自动生成算法，该算法利用输入/输出关系分析和原子组件测试用例。他们的方法显著减少了人工生成预期结果这一耗时且容易出错的工作。

端到端测试。这类测试通过评估整个机器人系统 [12]，验证系统是否已正确构建。在此级别，所有节点会一同运行，并在类似真实运行的条件下接受测试 [34]。端到端测试是部署前最高的验证级别；它对应于检查V模型左侧的系统架构。在ROS系统中，这类测试涵盖完整的软件和所有硬件部分，并在受控场景中进行评估，以确认系统行为符合预期。它们通常会自动执行，因此开发人员可以反复测试逼真的行为，并发现只有在所有节点交互时才会出现的问题 [60]。

已有多项研究探讨了机器人学中的端到端测试。Erich 等人 [60] 提出了一个开源框架，可对协作机器人应用进行自动化测试，并在与持续集成流水线集成的同时评估系统行为。Breitenhuber [34] 提出了一个适用于 ROS 的行为驱动框架，该框架支持测试优先开发，引入行为规范，并支持自动化端到端测试。Cardoso 等人 [41] 提出了一种验证模块化机器人系统的方法，并指导对单个组件进行测试，以增强对完整系统的信心。Cossell [50] 展示了如何通过验证多智能体平台中的软件环境和配置来改善系统设置并简化故障排查。

验收测试。这些测试通过检查机器人系统是否满足预定义需求 [12, 175]，来验证所构建的系统是否正确。测试会根据特定的运行场景评估整个系统，该场景由多项功能组成，以确认机器人能够呈现用户或利益相关者所期望的行为。与上一个级别一样，评估对象是完整系统，包括外部来源的输入和输出，例如障碍物或传感器噪声。验收测试体现了V模型左侧的验证活动，即根据需求 [55, 217]检查系统。此阶段需要用户或客户的确认，并确保系统适用于其预期用途。

已有多项研究探讨了机器人学中的验收测试。Estivill-Castro 等人 [62] 采用简化的模型-视图-控制器（MVC）模式，在剥离图形用户界面的仿真环境中测试完整的机器人行为，从而能够高效评估复杂任务，有时甚至比在现实环境中更快。Kramer 和 Murphy [130] 提出了一种用于无人地面车辆客户验收的自动化流程，该流程在标准化环境中长时间测试车辆的所有能力，并发现了大量失效。Nguyen 等人 [175] 进一步推进了行为驱动开发，加入计时和系统行为建模，从而能够创建特定的验收标准，并通过仿真自动进行检查。Santos 等人 [55, 217] 提出将业务需求转化为可执行场景，证明了该方法能够可靠地检测故障，并提升工业机器人系统的软件质量。

运行示例对于配送任务（参见表1，需求10），单元测试可以针对单个节点，例如速度控制器或障碍物检测组件，利用模拟传感器输入验证其独立运行时的行为。随后，集成测试可以组合多个节点，例如导航、避障和轨迹跟踪，以确保它们在机器人规划和执行路径时能够正确协作。接下来，端到端测试可以运行整个系统，观察机器人从厨房驶向顾客餐桌，同时避开障碍物并遵守安全约束。最后，验收测试会将机器人置于真实的餐厅环境中，设置动态障碍物并模拟与人的互动，以测试整个系统。

4.3 测试类型

测试类型定义了测试内容，并赋予测试以意义 [103], 无论测试针对的是功能正确性、性能，还是安全性和可靠性等其他质量属性。也就是说，测试分为功能测试和非功能测试。功能测试针对系统的功能，即验证系统的运行属性或一般属性。非功能测试针对系统在不同运行条件下的质量 [17, 88, 101], , 并分为两类：质量，即关注系统的质量；约束，即关注系统必须遵守的外部法规或安全标准等 [79]。

4.3.1 功能测试。 功能测试评估系统特定功能的完整性和正确性。因此，功能测试源自系统的功能需求，旨在验证每个组件是否符合这些需求。在机器人学中，功能测试通常侧重于确认机器人能否正确执行其设计要完成的任务或动作。也就是说，重点评估动态环境中的避障和导航性能 [183], 移动机器人的路径跟随精度和运动稳定性 [6,46, 111], 以及工业机器人中用于视觉反馈控制的感知系统 [129]。此外，功能测试还评估末端执行器能力，包括机械臂系统的抓取力、可重复性和循环时间 [63, 127, 141] 。在联网机器人应用中，功能测试包括验证分布式控制场景下的通信可靠性和延迟 [221]。

随着机器人系统的快速发展，人们致力于定义功能测试，旨在使不同应用领域的测试具有可比性并实现标准化 [90, 105, 177]。这包括建立多足机器人敏捷性基准 [58], 、无人机导航与障碍物评估标准 [275], , 以及将位姿估计误差与操作任务成功联系起来的基于任务的性能指标 [90]。总之，机器人学中的功能测试意味着确定功能正确性，并结合感知、控制和环境交互。

运行示例功能测试可以评估导航与障碍物规避能力，例如验证 1 号表中的需求 10 。测试目标是检验机器人能否在无碰撞的情况下到达目的地，同时保持定位精度，并适应其内部地图与物理布局之间的环境变化。功能正确性的验证可以依据 Norton 等人的标准化流程进行 [177]。机器人将在包含可配置静态和动态障碍物的有界环境中，反复在位置 A 和 B 之间导航。通过外部观测记录性能指标（例如任务成功率、平均通行时间和碰撞次数）。应用于我们的餐厅机器人时，我们设定两个位置，例如厨房（A）和 3 号桌（B），并添加椅子或移动的人等障碍物，以模拟现实情况。机器人将从 A 导航至 B，并在遇到意外障碍物时更新地图。通过测量任务完成时间、多次重复任务的成功率以及轨迹偏差，来判定功能正确性。

4.3.2 非功能测试。非功能测试关注的是与功能没有直接关系的方面，而是系统运行的条件和质量属性，例如性能约束和可用性期望 [103]。文献中对这一术语并无精确定义，且它通常涵盖定义不充分的功能需求。为解决这一问题，可将非功能需求进一步划分为质量属性和约束 [79]。前者涉及系统在不同条件下性能的可测量属性，例如可用性或资源利用率；后者涉及是否符合由利益相关者、监管机构或行业标准设定的外部规则或法规。

质量测试。质量测试并非评估系统行为是否正确，而是评估系统在各种运行条件下的表现。它依据既定的质量要求 [21]，评估负载下的性能、可扩展性和兼容性等质量特性。其目标是确定系统在工作负载或环境因素发生变化时的行为。例如，测试可以关注计算负载增加时响应时间的变化，以及性能是否仍处于可接受的范围内。质量测试能够证明软件在一系列预期条件下可以稳定、高效地运行。

可用性测试 评估用户使用系统的体验，包括系统是否直观、高效。可用性对人机交互尤其重要，因为界面的质量会影响用户接受度和任务表现。Golchinfar 等人 [80] 研究了在特殊情况下远程控制服务机器人的图形用户界面的可用性，并报告了相关经验教训，为设计能够改善远程人工干预的界面提供参考。Casiddu 等人 [43] 提出了机器人可用性测试的设计指南，涉及参与者招募、样本量、会话时长、侵入性以及测试环境的影响等方面。Moon 和 Kim[167] 为面向老年人的运动辅助机器人制定了定量可用性评估标准，并结合访谈和实证评估来衡量交互质量。

性能测试评估系统在典型或预期负载下运行时，执行预期功能的效率和可靠性 [98]。测试可考察不同运行条件对系统性能的影响，例如响应能力、资源利用率和容量 [103]。Wienke 和 Wrede [267] 提出了一个框架，用于指定、执行和分析针对机器人系统各个组件的性能测试。被测组件会集成到受控测试环境中，该环境生成输入数据和负载条件，同时监控性能指标。也就是说，测试会测量系统响应用户请求的速度、资源消耗如何随并发操作而变化，或系统性能是否会在峰值需求条件下降低。需要明确的是，准确性属于功能需求，而响应能力和负载处理则属于与性能相关的非功能需求。在系统级，Okwu 等人 [180] 通过测量自主割草机在不同草高下完成任务所需的时间来开展性能测试，展示环境因素如何影响机器人性能。同样，Gertman 等人 [77] 提出了一种评估人机团队性能的方法。根据所考虑的工作负荷强度 [98], , 性能测试通常会细分为更具体的测试类型，例如负载测试、压力测试、耐久性测试和可扩展性测试；本文将这些测试视为独立的测试类型。

负载测试 用于评估不同负载下的性能，负载通常从低负载和正常负载到峰值负载不等 [98]。其目标是确定临界点，即系统性能在真实的工作负载条件下开始变得不稳定且无响应的时刻。例如，确定在发生失效之前，系统最多能够处理多少并发用户、进程或传感器输入。我们没有找到明确针对机器人领域负载测试的论文。

压力测试 用于评估系统在负载超过预期峰值，或可用资源（例如内存、处理器、磁盘）低于规定的最低要求时的性能

，以评估系统在极端条件下的表现 [21, 98]。压力测试会将系统推至绝对极限，并评估其在超过临界点后的表现。Collet 等人[48] 介绍了 RobTest，这是一款通过最大化 CPU 负载对单臂机器人进行压力测试的工具。软件系统的压力测试还包括容量测试 [98], ，用于评估系统处理海量数据时的性能，或数据存储接近容量上限时的性能。然而，我们没有发现针对机器人领域容量测试的论文。

耐久性测试 用于评估系统在长时间连续运行期间维持性能和可靠性的能力 [98]。其目标是检测资源泄漏或性能逐渐下降等问题，这些问题在较短的测试中可能不会出现。在机器人学中，耐久性测试通常通过在较长时间内持续运行某个特定子系统（如感知、导航或通信），或整个系统来进行 [11, 36]。Ball 等人 [15]进行了为期数周的大规模现场测试，以评估障碍物检测和导航系统在真实条件下的鲁棒性。Babić 等人 [13] 开发了一种车辆在环测试环境，以便对自动驾驶系统进行长期测试，从而无需承受实地部署带来的安全和成本约束，便可持续评估自主性能。

容量测试 （也称为可扩展性测试）用于评估系统在未来可能需要支持的条件下的性能。这包括评估为支持预期的未来负载，需要增加多少资源（例如内存、磁盘容量、网络带宽） [98]。它可确保系统能够应对不断增加的负载，例如更大的数据量、更高的用户负载或更多的计算需求，同时不影响未来的性能。机器人学中的容量测试通常关注软件能否随着机器人数量的增加而扩展。Humphrey 等人 [96] 开发了一个面向人类—多机器人系统的可扩展接口，重点研究机器人数量增加对工作负荷、态势感知和机器人使用情况的影响。Hamann 和 Reina [86]提出了一个多单元系统的通用可扩展性模型，并将其应用于多机器人系统。

资源利用率测试用于评估各种运行条件下计算资源和能源资源的使用效率。它通常会测量 CPU 和图形处理器的负载、内存使用量及网络带宽，以识别瓶颈，确保资源在运行时得到有效分配。这些测量有助于验证软件系统是否满足性能和效率需求，同时避免过度消耗资源。然而，在机器人学中，资源利用率测试可确保即使在计算复杂的任务期间，也能满足实时约束。

Maruyama 等人 [154] 探讨了 ROS 2 中的性能和资源利用率方面，并着重指出确保可预测的时序行为所面临的挑战。同样，Bédard 等人 [22] 将 LTTng 等工具集成到 ROS 环境中，以便在测试期间监控 CPU 和内存使用情况。Wienke 等人 [266] 提出了一种领域专用语言（参见第 5.1 节中的基于模型的测试），用于系统地指定针对各个机器人组件的性能和资源利用率测试，从而支持参数化测试用例，以捕捉不同执行阶段对 CPU、内存和网络的需求。与传统软件相比，能源消耗是机器人软件的一项关键资源。Swanborn 和 Malavolta [241] 开发了 Robot Runner，这是一种用于自动开展实验以评估能源消耗以及 CPU/内存使用情况的工具。Touzout 等人 [250] 通过将差速驱动机器人的能耗模型集成到 ROS/Gazebo 仿真环境中对其进行了扩展，从而改进了监控和能源管理。Gadaleta 等人 [69]分析了高负载工业机器人的能源消耗，并测量其在不同负载条件下执行任务时的能耗，以验证虚拟原型。软件标准提及的一种特殊资源利用率测试是内存管理测试，其重点是根据内存使用量进行性能评估 [98]。不过，我们没有发现报告内存管理测试的机器人学论文。

可靠性测试 评估系统能否在变化的条件下长期持续执行其预期功能。其目标是发现可能

在短期运行中不会出现的失效或意外行为，重点关注稳定性、可信性以及失效频率。与强调持续运行及随时间推移的总体性能的耐久性测试不同，可靠性测试衡量平均失效前时间或平均故障间隔时间等指标，评估系统在规定条件下于指定时段内正确运行的可能性 [98]。在机器人学中，可靠性测试通常包括长期仿真、场景的重复执行以及真实世界实验，并常辅以系统性故障注入，以评估系统在异常条件下的鲁棒性 [89, 257]。Tang 等人 [244] 开发了专用测试平台，用于评估自动驾驶车辆中的实时预测诊断、健康管理以及自动应急管理技术，从而实现系统性的可靠性评估。Deng 等人 [53] 提出了一种基于时间自动机的方法，用于记录自动导引车中与可靠性相关的事件发生情况，从而提供系统可信性的定量度量。同样，Hereau 等人 [89] 为水下机器人引入了一种容错控制架构，其中不良事件由容错模块检测并管理，该模块要么恢复系统，要么向任务管理器报告，从而提高整体系统可靠性。

安全测试 用于评估系统是否存在可能危及完整性或机密性的漏洞或缺陷。其目标是识别可能导致未授权访问、篡改或操作系统及其数据的漏洞 [98]。在机器人学领域，由于机器人需要与物理环境、网络及人类操作员交互，安全测试尤为关键。Fernandez 等人 [64] 采用椭圆曲线集成加密方案，保护无人机通信系统中的数据传输免受恶意攻击。Reithner 等人[203] 对移动机器人进行了渗透测试，发现了可能同时危及机器人运行和网络安全性的漏洞。安全测试还针对 ROS 等中间件；例如，Sim 等人 [231]开发了 SMINER，用于识别 ROS 中因行为不受限制或实现错误而产生的漏洞。除工具和框架之外，机器人学中的安全测试还受益于结构化的工程实践。Mayoral-Vilches 等人 [157] 推出了 SROS2，这是一套适用于 ROS 2 的实用网络安全工具和库，可帮助开发人员为机器人系统添加身份验证、授权和加密功能。该工具简化了 ROS 2 计算图的内省、身份与权限证书颁发机构的生成与部署，以及将 ROS 2 安全配置转换为 DDS 权限的过程。他们提出了一种基于 DevSecOps 的六步方法论，用于保护 ROS 2 计算图，涵盖建模、工件生成和持续监控。

约束测试。测试用于评估系统是否遵守外部法规，或利益相关者、监管机构或行业标准 [98] 规定的需求。约束可涉及各种话题，包括安全法规、数据隐私与保护、保险政策和法律。

合规测试 旨在验证系统是否遵守特定标准、法规或法律要求，也称为符合性测试或法规测试。在机器人学领域，由于缺乏广泛采用的领域专用标准，系统通常必须遵守适用于其运行环境的法规。例如，自动驾驶车辆必须遵守交通法规和安全需求 [26]。Bensalem 等人 [23] 提出，针对由经仪器化的系统生成的执行轨迹，依据形式化规约检查其合规性，以实现实时应用程序符合性的自动化测试。Sohail 等人 [234] 结合基于仿真的测试和基于属性的测试，验证机器人是否符合标准，并将机器人标准中定义的需求形式化，自动执行这些需求，将其转化为可复用的仿真场景。这使得在不同机器人平台上部分自动化符合性测试成为可能，并能发现此前未检测到的软件和驱动程序缺陷。

安全测试着重确保系统在规定条件下不会造成伤害、对人类造成伤害或导致财产损失。安全测试可视为合规测试的一个子集，但与合规测试不同，它往往缺乏普遍适用且明确定义的规则，

因此，其评估取决于具体情境。在机器人学中，当人类与机器人共处于运行环境中，或在协作任务中与机器人密切互动时，安全性尤为重要 [78, 93]。Gleirscher 等人 [78]为协作制造开发了一种安全控制器，结合综合、形式化验证和测试，以确保安全的人机交互。Huck 等人[93] 将危险分析形式化为搜索问题，并提出基于仿真的策略，以生成测试用例，识别由人类意外动作引发的危险行为。针对足式机器人，Weng 等人 [264] 提出对场景进行采样，以刻画整体安全性能，并在各种运行场景下系统地测试潜在的失效模式。

运行示例为了评估餐厅机器人的资源（例如 CPU）利用率，我们可以使用 Wienke 等人的框架。 [266]借助他们的领域专用语言，开发人员可以对重复的路径规划请求或模拟的摄像头数据流进行建模，同时由框架监控不同输入速率或环境复杂度下的 CPU 和内存使用量。例如，导航栈应将 CPU 利用率维持在规定阈值以下，以避免运动控制或任务管理出现延迟。该框架会生成合成测试输入，模拟来自激光传感器的动态障碍物数据，并测量导航算法对这些数据的处理方式。

4.4 测试活动

明确了测试环境、测试级别和测试类型的含义之后，下一步就是以有意义的方式对它们进行组织，为完整的测试工作提供结构，明确测试什么、如何测试以及何时测试 [176]。测试活动是在整个测试计划中执行测试所采取的步骤。无论测试计划或策略如何，这些步骤都始终存在，用于定义测试的执行方式。我们不会详细讨论测试过程、测试计划和测试策略，因为它们因项目或组织而异，但会提供直观的理解。测试过程和测试计划为组织测试活动提供框架和指导。测试计划是一份详细文档，定义测试目标（通过测试应达成什么）、实现这些目标所需的策略和资源，以及执行测试的时间安排，从而协调所有测试工作。测试过程则由一系列系统化的测试活动组成，用于实现测试目标 [98]。测试策略是对组织或项目层面如何开展测试的高层级描述。它定义了指导所有测试工作的原则、通用方法和规则。相关研究会考虑优化方法，旨在确保测试项得到全面覆盖的同时，将总测试时间降至最低，而且通常还要应对资源约束 [81, 143, 245]。

需求分析与测试规划。此活动涉及分析系统需求、制定测试目标并选择适当的测试类型。此阶段结束时，将形成一份详细计划，明确测试系统的哪些部分、如何测试，以及测试依据哪些需求进行。

分析需求 识别与测试相关的所有功能需求和非功能需求。将对这些需求以及其他相关工件进行详细检查，例如设计规范、行为模型和源代码。它们共同构成测试依据，据此制定测试目标并确定测试项——即要测试的具体单元、组件或子系统。此步骤可确保后续测试活动系统地与预期的系统行为和质量特性保持一致。

测试规划 会根据需求分析，针对每个测试目标和其中的每项内容，定义测试条件（即应测试的内容）和测试类型——5 [101, 103]。由于通常难以对整个系统进行穷尽式测试，测试规划还包括确定优先级，明确哪些组件需要进行更全面的评估。最后，此阶段会确定最适合实现既定测试目标的测试类型（参见第 4.3节），并为后续的测试设计与执行制定结构化路线图。

测试设计与实现。此阶段会运用适当的测试设计技术，根据已定义的测试条件和测试类型，系统地推导或选择测试用例。在此过程中，测试条件会转化为具体且可执行的测试用例。设计测试套件（即测试用例的完整集合）时，会设定覆盖目标，以确保对被测系统进行充分验证。测试覆盖率是指测试套件对根据测试依据和测试条件得出的覆盖项进行测试的程度。覆盖范围可包括基于规格说明的覆盖（例如需求覆盖率）、结构覆盖率（例如语句、分支或条件覆盖率），以及对测试数据变化和执行环境的覆盖。测试用例与测试依据之间会保持可追溯性，以支持覆盖率分析和影响评估。需要注意的是，覆盖准则可作为衡量测试完整性的指标，但其本身并不能保证缺陷检测有效性。此阶段的产出是详细的测试套件，以及已定义的覆盖准则和可追溯性信息，为系统化的测试执行与评估奠定基础。

测试环境搭建与执行。测试套件设计完成后，将在受控测试环境中执行，该环境应符合每个测试用例定义的前置条件。测试环境可能由基于仿真的配置或真实环境配置组成。具体选择取决于多种因素，例如测试目标以及与执行相关的风险。

执行期间，会记录相关数据，例如性能指标和系统状态，以供分析。在机器人学领域，持续集成和持续部署流水线并不常见。不过，人们仍在采取改进措施，例如，Swanborn 和 Malavolta [241] 引入了 Robot Runner，这是一种旨在简化基于测量的实验执行的工具，可自动完成仿真机器人和实体机器人的实验搭建、执行与复现。**测试评估**。测试执行阶段结束后，会对收集的数据进行分析，并与预期结果进行比较，以确定每个测试是通过还是失败。该评估用于判断观察到的行为是否符合规定的需求，以及既定的测试目标是否已经达成。然而，评估的不仅是单个测试结果，还会对测试覆盖率进行全面评估，以确保测试套件覆盖了所有需求。评估阶段会反馈系统是否满足需求；偏差或失效可能表明存在设计缺陷、实现错误或测试不充分。如果检测到问题，则需要调整代码或测试套件。对代码库所做的任何更改都会触发回归测试，以确保这些更改不会引入新的故障。

运行示例我们以需求10（参见 3号桌 1）为测试依据，说明各项活动。首先，**需求分析**决定了测试的目的以及涉及哪些组件。对于需求10，测试目标是验证机器人能够导航至目标位置，同时避免碰撞并遵守安全速度限制。相应的测试项是组件控制层中的运动规划与控制和同步定位与地图构建组件。根据测试目标，可以推导出代表需求10可测试方面的测试条件，包括：(i) 计算通往目标的无碰撞路径，(ii) 检测并避开动态障碍物，(iii) 将最大速度限制在0.8m/s，(iv) 在距目标的容差距离内停下机器人。其次，**基于测试条件和目标进行的规划**将得出恰当的测试类型。对于需求10，这包括用于验证任务是否正确执行的功能测试、用于测量通行时间和速度的性能测试，以及用于确保符合速度和碰撞约束的安全测试。随后，将设计并实施具体的测试用例 。一个测试用例如下：

前置条件：机器人已启动，在厨房区域完成定位，且餐厅地图已加载至导航栈。测试输入：一条导航命令，要求机器人前往餐桌3，命令以该餐桌在餐厅中的坐标形式提供。预期结果：机器人规划出一条通往餐桌3的无碰撞路径，在导航过程中避开出现的动态障碍物，且速度始终不超过0.8m/s。后置条件：机器人到达餐桌3，在距餐桌0.3 m范围内停下，并向任务管理器发送任务完成信号。在**测试环境搭建与执行**期间，测试将在真实餐厅或仿真环境中执行。对于我们的示例，需要搭建一个餐厅的 Gazebo 仿真环境，并配备日志记录与监控工具，以采集机器人的位置、速度和任务状态。最后，在**测试评估**中，将把测试结果（例如，轨迹/速度日志和配送完成时间戳）与预期结果进行比较。碰撞、超速或漏报配送确认等偏差将记录为失效，并通过回归测试验证修正是否有效。

5 测试设计技术

我们已经讨论了如何组织测试过程，以及如何赋予每项测试意义。现在，我们将探讨如何从不同的信息来源系统地推导测试用例。这些测试设计技术分为：黑盒测试，也称为基于规约的技术；白盒测试，也称为基于结构的测试；以及基于经验的测试 [98, 121]。

我们发现，大多数论文关注黑盒测试（47 篇），而白盒测试几乎没有涉及（2 篇）。基于现场的测试所涉及的论文数量位居第二（41 篇），其次是模糊测试（8 篇），最后是基于经验的测试（1 篇）。在黑盒测试技术中，基于模型的测试（21 篇）和基于场景的测试（16 篇）最为突出。图6 展示了这一分布。

5.1 黑盒测试

在黑盒测试中，系统被视为一个黑盒；也就是说，测试用例是根据系统外部可观察到的行为或规范中描述的行为推导出来的，而非依据其内部结构。行为可以通过模型、需求或其他形式来描述，并据此推导测试用例。在机器人学中，黑盒测试被广泛应用，例如用于测试系统的鲁棒性，方法是

通过修改有效请求序列 [197], 生成测试用例，或使用基于 Web 的黑盒测试框架测试机器人软件的组件 [113]。

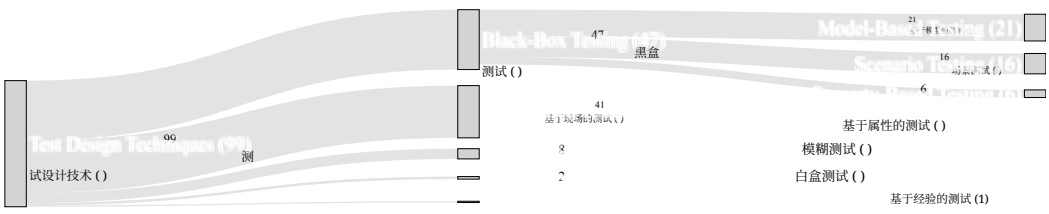


图 6. 报告了相关测试设计技术的机器人学论文数量

边界值分析 侧重于识别划分边缘或边界处的错误（通常通过等价类划分得出）。针对每个边界，测试会关注紧邻边缘两侧的值——既包括等价类内的值，也包括刚好位于等价类之外的值 [98, 101]。我们没有发现任何报告机器人学中边界值分析的论文。

分类树方法 使用树形表示将测试输入（如等价类划分）分组为互不重叠的类别。每个分类代表需要测试的相关方面，例如输入类型或系统状态 [98]。我们没有找到任何报告在机器人学中采用该方法的论文。

组合测试 获取有意义且规模可控的测试用例子集，通常用于测试具有离散值的大量参数之间的交互。选择具有特定值的参数组合可以减少测试用例的数量。组合被定义为参数 P 及其可取的值 V。测试覆盖率通过选择 P-V 对的子集来确定。存在多种推导 P-V 组合的技术，例如所有不重复的组合、成对组合、至少一次取值为 V 的 P-V 对集合（也称为每种选择测试），或通过将所有参数固定为“基本”值，然后一次改变一个参数的值，将其设置为非基本有效值而得到的 P-V 对集合（称为基本选择测试） [98]。Klück 等人 [128] 使用组合测试设计技术，通过本体描述环境，为高级驾驶辅助系统创建并管理场景。

决策表测试通过评估条件组合来确定不同的结果，对逻辑关系尤为有效。这些表通过定义系统条件及相应动作来构建。每一列代表一条决策规则，即一种独特的条件组合及其关联动作。在有限条目决策表中，条件和动作的取值通常为布尔值。相反，扩展条目决策表的条件和动作可以取多个值，例如数值范围、等价划分或离散值 [98, 101]。我们未发现任何论文报告机器人学中的决策表测试。

基于模型的测试（基于模型的测试）。基于模型的测试系统地从定义系统所需或预期行为的行为模型中推导测试用例，不应将其与环境模型或仿真模型相混淆， [98, 101]。这些模型抽象了系统的不同方面，针对整个系统或各个组件 [61, 198], , 或在不同情境中刻画行为，例如人机交互 [163], , 并通过探索模型的状态与状态转移来支持系统化的测试用例生成 [117, 163]。

常见的表示方式可以是形式化或半形式化的，包括状态机、自动机或领域特定语言 [258]，用于描述预期行为和交互 [91]。状态转换测试是基于模型的测试的一种特殊情况，它将系统建模为状态机，描述所有相关状态和状态转移 [98, 101]。输入会引起系统状态的变化，而测试用例旨在覆盖状态之间有效和无效的状态转移。测试用例通常表示为一系列输入/事件，由此产生一系列状态变化（以及必要时的动作）。

基于模型的测试已广泛应用于机器人学，尤其是因为行为模型利用复杂的交互描述，关注安全性、鲁棒性分析、交互（人机或多智能体）、边界情况探索和覆盖率等不同方面。Micskei 等人[163]提出了上下文模型，用于描述复杂情境下的行为，以保障自主系统控制软件的安全性和鲁棒性，并支持生成上下文感知场景，进而从中推导出测试用例。Zendel 等人 [277] 提出了 VITRO 框架，通过根据领域描述自动创建三维场景，并据此生成测试用例，来评估计算机视觉解决方案的鲁棒性。

对于协作任务，模型用于定义交互行为并生成测试用例，例如在人机交互 [260], 或自动驾驶 [218]中。彩色Petri网已被用于对交互和系统动力学进行建模，从而支持对协作

行为 [139, 213]的验证。测试用例生成通过Petri网的覆盖准则和进化算法进行引导，以最大化交互次数，同时尽量减少测试用例数量 [213]。在人机交互中，信念-愿望-意图智能体模型模拟智能且自适应的机器人，提供理性主体性和用于规划的推理机制，以支持逼真的测试生成 [9]。

领域特定语言 [258], 或其他工具和框架支持系统化生成测试用例，包括边界情况识别。例如，图形化领域专用语言被用于将模型指定为由位置和状态转移构成的图，并为其标注概率和事件，从而能够在仿真环境中自动生成和执行测试用例 [198]。TestIt框架支持对ROS软件包进行集成测试，并提供与 Uppaal 等模型检测工具集成的接口 [116, 117]。它允许根据行为树或其他规范生成模型，并为依赖项创建占位组件，从而能够测试复杂软件。

Núñez 等人 [178] 通过引入 ptockLTS 模型并定义一套全面的实现关系，提出了针对概率型循环机器人控制软件的形式化测试理论，使得在不同可观测性假设下能够进行可靠、系统的基于模型的测试。扩展有限状态机已被用于建模环境行为和环境因素。测试用例首先根据抽象世界模型上的图覆盖准则生成，然后通过输入空间划分将其转换为可执行的测试用例 [8]。

后续研究旨在开发工具和框架，或将持续集成融入基于模型的测试工作流程，以自动化测试生成与执行 [91, 119, 122, 153, 169, 261, 263]。

运行示例对于~~我们的餐厅机器人，我们可以使用基于模型的测试和VITRO框架评估~~感知系统，尤其是需求3和8 [277]。首先，我们需要定义一个领域模型来表示餐厅环境，其中包括相关物体（例如桌子、人手和托盘）。为每个物体指定视觉属性（例如形状、尺寸、颜色和纹理），并关联相应的三维模型。VITRO将变化的视觉条件（如阴影、反射以及顶灯造成的眩光）称为“视觉关键因素”，这些条件可以从提供的目录中选取。基于该领域模型，VITRO可以自动生成三维餐厅场景并渲染合成图像，系统地改变物体配置和照明，以实现可测量的领域覆盖率。例如，它可以生成手被其他物体部分遮挡，或运动模糊影响手势的场景。每张渲染图像都会配有自动生成的真实标注，例如“盘子”和“手”的物体掩码与边界框。通过聚类和均匀分布采样，VITRO可以减少冗余测试用例。随后，将这些生成的数据集输入感知组件，以评估其在变化条件下的性能。自动生成的真实值可用于评估功能指标（例如精确率和召回率）以及鲁棒性指标（例如对照明变化或遮挡的敏感度）。在实践中，VITRO的渲染设置会模拟机器人的相机参数（视场、分辨率、镜头畸变），以确保输入条件真实可信。

场景测试。该方法通过逼真的（通常是端到端的）交互序列来评估系统行为，而不是孤立的输入/输出对 [98]。场景涵盖系统可能遇到的真实情境 [42, 172, 238]。场景描述一系列事件中的预期行为，既包括常规（“主要”场景），也包括意外或故障（“替代”场景）情况。在机器人学中，基于场景的测试得到了广泛应用，因为它能够轻松指定顺序交互、环境背景和动态条件。领域特定语言通常用于建模环境并定义多样的测试场景，从而能够以结构化格式指定复杂场景和环境条件 [163, 200, 201]。此类

领域特定语言能够为基于仿真的测试系统地、可重复地生成测试场景。这在机器人学和自动驾驶等领域尤其有用，因为这些领域中的系统行为取决于一系列交互和上下文，而非孤立的测试用例 [201]。

机器人场景测试的应用范围广泛，从验证操作和导航行为，到评估不确定性下的鲁棒性。例如，STAMINA 项目使用六种典型场景评估机器人配套系统，以考察其在真实条件下的性能 [131]。场景测试还被用于测试基于预测的决策算法在传感器噪声和时序扰动下的鲁棒性 [242], ，根据行车记录仪录像重建碰撞场景，以测试自动驾驶算法 [20], ，以及通过三个受控场景验证无人机在固定或移动平台上的着陆策略 [85]。类似地，Fornasier 等人 [68]推出了 VINSEval，这是一种基于仿真的框架，利用参数化场景，并对传感器噪声、照明和环境条件进行受控变化，从而系统地评估视觉惯性导航系统（VINS）。

多种工具支持场景生成与执行。AmbieGen 使用进化算法进行基于搜索的场景生成，以识别关键场景 [94, 95]。为了测试涉及人类与机器人交互的场景，可部署模型驱动框架：这类框架会根据配置文件自动生成形式化模型和属性，并在真实或仿真环境中运行统计模型检查实验 [136]。对于自动驾驶车辆，可以使用模仿学习代理引导自适应搜索，迭代训练模仿代理以模拟真实代理，并生成对场景性能越来越准确的预测 [170]。Ortega 等人 [182] 推出了用于移动机器人基于仿真的测试的可组合且可执行的场景，以及支持开发人员对其进行规范、复用和执行的工具。这些场景允许在不修改现有模型的情况下添加新语义，例如动态元素（如门）或变化的任务规范，从而创建内容丰富且易于管理的场景。Kang 等人 [115]提出了 BTScenario，这是一款面向自动驾驶系统的场景规范与测试用例生成工具。BTScenario 支持对地图对象（如道路和交叉路口）的空间布局进行高层描述，并采用基于行为树的方式规范参与者（包括汽车和行人）的时序行为。此外，使用扩展UML时序图表达场景模型的框架还可开展结构化的鲁棒性和功能安全评估，包括详细的轨迹比较和覆盖率指标 [163]。

运行示例我们可以通过场景测试和 AmbieGen 框架，评估餐厅机器人的导航和避障能力（需求10，参见表 1） [95]。在 AmbieGen 中，餐厅环境将由其尺寸、障碍物尺寸的最小值和最大值以及位置参数化。它会将这些环境参数编码为基于搜索的优化过程中的个体，并逐步演化出可能揭示导航故障的场景。每个生成的场景都会被转换为 Player/Stage 仿真，机器人将在其中尝试将订单从厨房送到目标餐桌。为了在特定环境中发现失效，我们可以基于 FloorPlan领域特定语言 [189]定义一个新的场景生成器。这将能够生成更真实的环境描述，更精细地控制环境生成（例如餐厅布局，或桌子和椅子的位姿与尺寸），并支持在 ROS/Gazebo 中运行场景。每次运行期间，系统都会将指标（例如路径完成度、碰撞事件和行驶时间）记录为 ROS日志。对于机器人被阻挡、发生碰撞或违反安全约束（例如速度超过 0.8 m/s）的场景，将赋予更高的适应度分数，以引导生成器寻找狭窄过道、部分受阻路径或人流密集等关键情形。导航或感知更新后，可以重放生成的场景，以验证改进效果。

基于属性的测试。基于属性的测试用于验证系统是否满足形式化指定的属性或不变量，而不是逐一检查输入/输出对 [65]。

在机器人学中，基于属性的测试已用于验证基于ROS的系统 [214]中的运行时不变量。典型属性包括约束，例如确保机器人不会离开指定的工作空间，或确保传感器值始终处于有效范围内。Santos 等人 [215] 介绍了 HAROS 框架，该框架提供了专用于 ROS 的基于属性的测试工作流程：从源代码中提取系统模型，将高层行为属性转换为由 Hypothesis 驱动的测试生成器，并通过运行时监视器自动检测反例消息轨迹。Robert 等人 [205] 提出了一种基于仿真、以属性为导向的测试方法，通过随机生成虚拟农田并采用基于不变量的预言机，在多种任务场景中揭示农业机器人的导航故障。Santos 等人 [216] 根据度量一阶时序规约自动生成测试用例，使高层行为描述能够驱动系统化的测试生成。基于属性的测试也被用于通过自动创建仿真测试场景来评估机器人的正确性。Sohail 等人 [233]提出了一个基于属性的仿真框架，用于验证单个及多个动作任务，并自动在不同条件和参数下生成测试并进行评估。蜕变测试是一种相关技术，它将基于属性的测试扩展到难以精确定义预言机的情形。Zhang 等人 [279] 提供了一个用于定义蜕变关系的框架。蜕变关系描述多个输入和输出之间的预期关系，可用于为路径规划算法构建预言机。

运行示例我们可以使用 HAROS 框架对基于 ROS 的软件进行基于属性的测试 [215]。属性将用 HAROS 属性语言表达，并通过 Hypothesis 库自动转换为可执行的测试模式。这样便可以验证诸如“发布在 `/cmd_vel` 上的机器人的速度绝不能超过 0.8 m/s”（参见表 1，需求10）或“每当发布一条 `delivery_started` 消息时，必须出现相应的 `delivery_completed` 或 `delivery_failed` 消息”（参见表 1，需求7）这样的属性。HAROS 会解析项目中的 ROS 软件包和启动文件，构建通信图，然后在仿真或运行时注入消息序列并监测主题值。Hypothesis 引擎会系统地改变消息的时序、顺序和内容，以探索不同的系统行为并揭示反例。

5.2 白盒测试

白盒测试也称为基于结构的测试，它利用对内部结构或实现的了解，通过分析源代码或其他内部表示来推导测试用例 [98]。白盒测试主要用于单元级或组件级测试，或低级集成测试 [88]。测试标准 [98, 101] 以覆盖率指标规定白盒测试，但并未说明如何生成具体的测试输入。实践中，测试输入通常由工程师通过代码检查和领域知识手动推导，或使用符号执行、约束求解以及随机或引导式输入生成等技术自动生成。在机器人学中，白盒测试可用于验证运动规划算法的正确性，确保所有控制路径都得到执行 [123]，；也可利用模块内部逻辑，在模块化机器人系统中检测分布式错误 [209]。

语句测试 导出测试用例，使所考虑的源代码部分中的每条可执行语句至少执行一次 [98]。对代码进行分析，以识别能够实现完整语句覆盖的输入；当某些语句无法被执行时，这也可以揭示死代码或不可达代码 [88, 98]。我们未发现任何报告机器人学中语句测试的论文。

分支测试 创建能够执行控制流图每一分支的测试套件 [88, 98]。为此，对控制流图进行分析，以识别触发控制流中每个分支的输入，从而可能揭示不可达分支。我们未发现任何报告机器人学中分支测试的论文。

判定测试创建测试用例，使所考虑的源代码部分中的每个判定都以所有可能结果（例如真和假）至少执行一次 [88, 98]。判定是控制流可以采取替代路径的任意位置，例如 if、while 或 case 语句中的条件。需要识别能够触发每种可能判定结果的输入。重点在于使每个判定结果至少被执行一次。因此，所得测试套件包含分支测试。我们未发现任何论文报告机器人学中的判定测试。

分支条件测试在判定测试的基础上加以扩展，重点在于评估各个原子条件，而不仅仅是整体判定结果 [88, 98]。例如，对于表达式 if (A or (B and C))，判定测试只考虑整个表达式的求值结果为真还是为假，而分支条件测试则要求每个原子条件（即 A、B 和 C）都独立地至少各取一次真值和假值。分支条件组合测试进一步要求执行原子条件取值的所有可能组合，从而对于 N 个条件产生 2^N 个测试用例 [88, 98]。修正条件判定覆盖通过找出判定内部所有能够独立影响其结果的原子条件组合，对此作了细化 [88, 98]。测试用例根据原子条件推导得出。我们未发现任何论文报告机器人学中基于分支条件的测试技术。

数据流测试 关注变量在源代码中的定义、使用和删除方式。通过分析控制流，识别定义（赋值）和使用（读取值，并将其用于计算或决策），进而推导出测试用例。针对每个定义-使用（DU）对，选择能够执行该DU路径的输入 [98]。路径的粒度可以有所不同：可以检查每个定义是否至少被一个测试用例覆盖（全定义）、定义是否流向计算使用（全计算使用）、是否流向谓词使用（全谓词使用），或者是否覆盖这两种使用（全使用）。全DU路径是最严格的变体，它会检查每个定义到其使用的所有可能路径 [98]。我们没有发现任何报告机器人学中数据流测试的论文。

5.3 基于经验的测试

这种测试设计技术依赖开发者的知识，而不是从模型、代码或其他规范中系统地推导测试用例。开发者会识别系统的相关方面并创建测试用例，也就是说，测试用例是以临时方式推导出来的 [88, 98]。

错误推测 根据对系统中错误类型和可能错误来源的了解来推导测试用例。测试用例可以源自经验、错误分类法或清单、缺陷或事件追踪记录，或其他知识库 [88, 98]。我们没有发现报告机器人学中错误推测的论文。

探索性测试 使开发人员能够即兴设计和调整测试，以发现系统可能导致失效的“隐藏属性” [103]。测试设计依赖开发人员的领域知识、关于系统行为的启发式方法，以及从以往测试结果中获得的见解。在机器人学中，探索性实验被提出作为受控测试的替代方案，因为不可预测的真实世界环境使得难以预先规定结果。Amigoni 和 Schiaffonati [7] 介绍了自主机器人学中的探索性实验，因为环境难以预测，开展传统的受控实验可能颇具挑战。

5.4 模糊测试

模糊测试 [165], 或称模糊测试，是一种基于大量随机或伪随机输入 [103]的测试用例生成技术。与系统化技术不同，模糊测试不需要明确规定预期的功能行为，而是通过观察程序在

受压输入条件下的反应来发现程序崩溃、超时、内存安全违规或其他 sanitizer 报告的错误。根据可用的系统知识和检测手段，模糊测试可应用于黑盒、白盒或灰盒测试，并通常利用反馈机制（例如代码覆盖率）来引导输入生成并提高故障发现率。

用于机器人学的模糊测试框架，例如 ROZZ [271] 和 RoboFuzz [124], 通过支持多维输入生成、分布式节点覆盖和时序变异策略，扩展了传统模糊测试。这些框架既可以针对低级软件故障（例如内存错误、崩溃），也可以针对更高级别的语义缺陷或信息物理缺陷（例如违反领域特定不变量或现实约束）。RROSFuzz 通过为不同版本或实现的 ROS软件包生成并变异输入，然后在 RViz 中比较其行为，测试机器人软件的一致性和鲁棒性 [255]。PHYS-FUZZ 将机器人尺寸、轨迹和撞击时间等物理语义纳入输入生成，从而有效识别仿真环境（如 Gazebo）中易发生碰撞或存在危险的导航行为 [270]。

一些研究专注于发现控制和行为逻辑中的故障。RVFuzzer 使用面向机器人车辆软件的控制引导模糊测试，通过变异控制参数，并将控制不稳定性作为反馈，揭示会导致不安全或不稳定运动的输入验证缺陷 [126]。类似地，Delgado等人 [52] 将基于语法的模糊测试应用于 ROS 的 SMACH 框架中的机器人行为，生成结构化的状态机输入，以识别编码、逻辑和配置错误。MDPFuzz 关注基于马尔可夫决策过程的模型所学习的决策策略（例如强化学习和基于规划的控制器）在意外或对抗性条件下的表现 [187]。它会变异初始环境状态，并观察由此产生的轨迹是否会导致不安全或异常结果。

在机器人学中，模糊测试对于识别复杂分布式系统中的鲁棒性和信息安全问题尤其有用。Wang等人 [256] 使用 RoboFuzz 进行定向模糊测试，注入符合机器人状态和环境、看似合理但有害的传感器值（例如伪造的距离读数），从而揭示控制逻辑中的漏洞，这些漏洞可能导致不安全行为。

5.5 基于现场的测试

基于现场的测试是一种测试设计技术，其测试用例在生产环境中执行，或源自生产环境。尽管现有测试标准尚未对此作出定义，但它在机器人学中发挥着核心作用，因为实验室环境无法完全重现真实的环境条件和交互。现场测试通常在端到端或验收层面进行，以评估功能需求和非功能需求；可分为体外测试和体内测试：体外测试在开发环境（内部）中执行，使用真实世界数据重现现场场景；体内测试则直接在生产环境中执行。测试可以在独立的系统实例上离线运行，从而降低运行风险；也可以在线运行于已部署的系统本身。体外测试和离线测试主要针对功能故障，而在线测试还涉及性能、信息安全和鲁棒性等非功能属性 [25]。在机器人学中，基于现场的测试服务于不同目标，包括旨在真实世界情境中发现意外行为和边界情况的探索性现场测试，以及用于评估机器人在生产环境中持续负载下长期性能和鲁棒性的耐久性现场测试；后者通常在集成级或系统级进行 [146, 181]。尽管基于现场的测试耗费资源，且需要完整构建系统，但它能够提供实验室测试或基于仿真的测试无法获得的见解 [76, 229]。基于现场的测试通常需要对软件进行检测配置并为此构建软件；这一做法已针对基于ROS的系统开展过研究 [40]。

在文献中，基于现场的测试通常以临时方式进行，用于评估真实环境条件下的系统性能、可靠性或组件行为 [59, 137, 276]。许多研究报告了机器人在现场的部署和评估，凸显了基于现场的测试对于验收测试和安全性的重要意义 [66, 107, 132]。研究对象涵盖无人机等不同类型的机器人 [51, 70, 162, 166, 239, 251], 无人地面车辆 [3, 44, 82, 211], 或人形机器人 [16, 132, 259]。基于现场的测试可应用于不同领域，例如农业 [29, 185, 272, 273, 280],重型作业 [84, 190, 243], 医疗保健 [145, 146, 156], 或太空 [18, 133, 265]。测试对象往往不只是单个机器人，还包括执行协作任务的多机器人系统 [49, 66, 118, 135, 188, 196]。

6 挑战与差距

尽管已有许多技术，机器人测试仍面临诸多挑战。一些综述论文明确提到了这些挑战，以下根据我们确定的五个类别进行总结。**仿真—现实差距**。如第4.1节简要讨论的，仿真—现实差距描述了仿真与真实世界在保真度方面的差异 [5]。虽然仿真在机器人学中发挥着关键作用，但创建能够达到与真实世界相同保真度的仿真既耗费成本又费时，而且往往难以实现 [247]。机器人仿真难以准确模拟真实世界物理特性、传感器误差、机械磨损和环境复杂性 [240]。摩擦、变形和不完善的传感器校准等因素会导致仿真性能与真实世界性能之间出现差异 [2]。此外，在仿真中对移动物体，尤其是人类（例如行人）的行为进行建模往往难以实现 [32, 112, 120]。

机器人学界为有效利用仿真付出了大量努力，我们在第4.1节的结果也印证了这一点。许多论文探讨了如何构建具备足够保真度的仿真环境。然而，针对不同机器人类型的标准仿真仍然缺失。**缺乏标准**。缺少标准和统一的测试流程，加上 ROS 等多种中间件 [202], YARP [161], 和 SORA [67], , 可能导致测试策略各自为政。机器人系统的应用范围广泛，从人机交互到自主配送无人机，这使得制定标准 [31, 32, 152, 195] 或用于统一验证系统的标准测试路线尤为复杂 [106, 108]。因此，不同研究的结果难以直接比较，整体可复现性也受到限制。为应对其中一些挑战，Caldas 等人 [40] 提出了相关指南，帮助开发人员和质量保证团队开发和测试机器人，尤其针对现场测试。

缺乏标准通常伴随着需求和系统覆盖率不足（例如代码覆盖率）。标准能够提供指南和结构，而覆盖率则必须明确界定，才能建立对系统的信任。尤其对于输入空间实际上无限的机器人系统，系统地覆盖边界情况十分重要 [1, 163], 。不同于通常具有确定性且测试用例可以明确定义的传统软件系统，机器人系统运行于动态且往往不可预测的环境中。此外，传感器噪声、执行器变异和人机交互使边界情况的识别变得复杂，且往往难以实现 [1, 163, 249]。例如，确定机器人对行人的正确行为仍然很困难 [32,112, 120, 152]。不系统的测试流程往往导致覆盖准则不完整或缺失，并依赖开发人员的领域知识，因此无法系统地覆盖边界情况 [163, 249]。

在机器人学中，测试设计技术（例如基于场景的测试）通过从特定场景中推导测试用例来解决这些问题。特别是，领域特定语言和模块化场景描述为系统地描述复杂场景提供了基础。尽管这是实现系统化覆盖率的第一步，但它们仍可能生成不现实的场景 [31, 32, 163, 247]。基于属性的测试则通过定义必须始终成立的属性，来应对无限输入空间带来的挑战

。这使我们能够抽象每个测试用例的细节，并聚焦于系统所需的不变量或安全需求。**测试自动化**。标准和覆盖率对于组织和指导测试十分重要，而测试自动化能够加快这一过程，且不一定需要以标准为基础。测试自动化主要分为两个步骤：（1）自动创建有意义的测试用例；（2）开发用于自动执行和评估的流水线。对于传统软件开发而言，持续集成与部署流水线已十分成熟；然而，在机器人系统开发中，这类流水线仍然相对落后 [2, 120, 163]。利用仿真进行自动化虽然威力强大，但也颇具挑战 [2]。自动化通常针对特定的模拟器和环境表示进行定制，因此难以在多个项目之间复用。此外，机器人学高度依赖基于现场的测试，使测试自动化几乎不可能实现，因为在物理环境中测试难以自动化，而且每天能够执行的场景数量非常有限 [106, 179, 195]。**可复现性与可扩展性**。可复现性与可扩展性是跨领域的普遍难题。由于机器人系统存在硬件非确定性、环境变化以及人为参与等因素，这些问题在该领域尤为突出。真实环境条件各不相同 [179, 195], ；即使输入名义上相同，硬件差异和传感器噪声也会引入非确定性 [2, 249]。此外，模拟器并非完全确定，在不同机器上会产生不同结果 [247]。总之，可复现性问题不仅存在于不同项目之间，也存在于同一机器人的不同版本之间，因为硬件上的细微偏差就可能导致行为不同。另一方面，物理环境限制了可扩展性，其规模自然难以扩大，而且每天只能执行数量有限的测试用例 [106, 179, 195]。仿真可以缓解这一问题，但运行大量仿真测试的计算成本很高，往往难以实现 [120]。此外，更多测试设计技术（例如基于场景的测试）并不能线性扩展，而多机器人系统可能导致协同场景出现组合爆炸 [163, 179, 195]。

预言机问题。确定测试的预期结果（即预言机问题）是软件测试中的一项普遍挑战，在机器人学中尤其困难 [19]。机器人测试虽然继承了传统软件测试中的这一问题，例如规范不完整和需求含糊，但现实世界的不确定性、物理交互和硬件变异性使问题更加复杂 [1]。因此，预测系统的正确响应变得困难，因为正确、可接受和错误的系统行为之间的界限日益模糊 [1, 4]。通常，多种响应都可能是正确的，它们以不同方式解决问题。为应对预言机问题，机器人测试高度依赖基于模型的测试、场景测试和基于属性的测试，这些方法也提供真实值。然而，开发模型仍然是一项复杂且具有挑战性的任务。

7 讨论

我们了解到，在机器人测试文献中，开发后期开展的系统级测试（例如端到端测试和验收测试）占据主导地位。具体而言，系统级黑盒技术（如基于场景的测试和基于属性的测试）最为常见。这一趋势也体现在仿真测试和基于现场的测试的广泛应用上；这两种测试方式都天然契合系统级评估或验收测试，因为它们测试的是作为一个整体的机器人，而非孤立的组件。文献中对开发早期阶段的测试（如单元测试和集成测试）的关注不足。过度依赖后期系统级测试可能会带来问题，因为这会使故障定位更加复杂。如果只在系统级检测到失效，往往难以判断根本原因究竟在于某个独立组件、组件间的交互，还是整体系统架构。这会增加调试工作量，并提高故障解决成本。未来研究应**将系统化的测试活动引入**

开发早期阶段，以符合既定的软件测试标准。具体而言，需要针对控制、感知等领域定制白盒测试技术，以便在单元测试和集成测试级别实现系统化覆盖。将此类技术整合到机器人中间件和开发框架中，有助于在整个开发过程中持续开展测试。

我们观察到，相较于功能正确性，研究更加重视非功能测试，例如鲁棒性和性能。在机器人学中，功能测试与非功能测试之间的界限往往变得模糊，因为安全性、鲁棒性和性能等属性通常被视为功能需求，而不是“可有可无”的质量属性。这一观察表明，将传统测试分类体系应用于机器人学时，可能需要进行调整。将鲁棒性和安全性视为非功能属性，可能导致对预期系统行为的规范描述不够充分，进而阻碍系统化的测试设计与评估。未来研究应致力于**更加清晰地规范和分类机器人系统需求**，从而实现更加系统化的测试设计，明确将功能预期与可测量的系统质量属性联系起来。

我们的综述进一步表明，机器人学中的测试流程通常是临时性的，并且由实验驱动。测试用例往往来自使用场景或领域知识，而不是系统化方法。边界情况通常依据开发人员的经验以及机器人的运行领域来识别。这些做法限制了测试的可重复性和可靠性，进而可能影响机器人的可信性。如果没有系统化的覆盖率或明确定义的测试准则，就很难评估系统在部署前是否已得到充分验证。此外，缺乏系统化测试和标准会阻碍不同研究之间的可复现性和可比性，从而可能对文献中实验结果的可信度产生负面影响。未来研究应**定义轻量化、系统化且能够感知资源的测试流程，并与既有的软件测试标准保持一致** 同时考虑机器人学中的约束，例如硬件可用性有限、测试时间受限以及计算资源有限 [81]。测试流程和报告实践的标准化将显著增强机器人学界内结果的可靠性和可比性。

最后，弥合仿真到现实差距仍然是一项关键挑战 [5]。虽然基于仿真的测试能够实现可扩展性并提供早期反馈，但基于现场的测试对于揭示现实世界的不确定性和未建模效应至关重要。**推进混合测试策略，系统地结合基于仿真的测试与基于现场的测试**，有助于提高人们对系统在真实运行条件下行为的信心。**进一步自动化测试生成、执行和评估**，并辅以统一的测试流程，将提升可扩展性、减少人工工作量并加快测试周期。

8 相关工作

现有文献综述仅涉及机器人系统测试的特定方面，例如形式化验证、仿真工具或安全保障。Luckcuck 等人 [144]对自主机器人系统的形式化规约与验证开展了文献综述。他们指出了这类系统面临的独特挑战，并评估了当前的最新进展。Santos 等人 [56]对机器人软件工程开展了映射研究，梳理了关键概念与实践。他们的研究强调，机器人软件工程需要更为系统化的方法，尤其是测试方法，因为这方面的研究代表性不足。然而，他们并未对机器人系统测试提供系统化概述，而是更侧重于一般的软件工程实践，而非测试。Song 等人 [235] 开展了一项工业界与学术界联合研究，以识别机器人测试的相关概念，弥合学术研究与工业需求之间的差距。Garousi 等人 [75] 对嵌入式软件系统领域的最新实践开展了系统性文献映射。他们回顾了 312 项原始研究，并根据其在测试技术、工具方面的贡献以及所揭示的产业界和学术界趋势进行分类。该研究

梳理了相关研究，并帮助从业者了解现有方法、避免测试工作重复，同时指出了嵌入式系统测试面临的开放挑战。Araujo 等人 [12]开展了首项针对机器人系统验证与确认技术的大规模系统性文献综述，分析了 195 项研究。他们发现，相关研究主要采用形式化模型（例如时序逻辑、状态机）和通用充分性度量，工具则主要集中于模型检查器和模拟器。该研究指出，领域专属度量指标、统一的基准测试以及工业证据方面仍存在不足。相比之下，我们的研究致力于将机器人测试文献映射到既有的软件测试标准和分类体系，提供一种强调对齐与覆盖率、而非以技术为中心的分析视角，从而揭示既有软件测试概念在机器人软件测试中的应用情况及其缺失之处。

9 结论

我们介绍了机器人测试领域的最新进展，并将其与软件测试理论联系起来。通过将理论映射到机器人学的实际情况，我们从软件测试知识体系中找出了相关技术，从而揭示了研究与实践中的不足。我们还描述了文献中报告的挑战，并探讨了目前应对这些挑战的方式。我们的分析显示，机器人测试通常在系统级进行，侧重于端到端测试或验收测试。单元测试和集成测试很少见，黑盒技术占主导地位，而白盒技术的应用有限。测试流程往往是临时制定的，严重依赖专家经验，而非标准化流程。这些发现表明，机器人学缺乏系统化且可重复的测试实践。为推动其发展，未来工作应优先建立标准、定义结构化测试流程，并缩小仿真与现实之间的差距；这一差距仍是确保结果可靠且可迁移的一项长期挑战。

致谢

部分经费由欧盟 Horizon 2020 项目 SESAME（编号 101017258）以及德国联邦研究、技术与航天部（BMFTR）通过德国机器人研究院（RIG）提供。

参考文献

[1] A. Afzal、C. L. Goues、M. Hilton 和 C. S. Timperley。2020年。机器人系统测试所面临挑战的研究。载于 *ICST*。[2] A. Afzal、D. S. Katz、C. Le Goues 和 C. S. Timperley。2021年。用于机器人测试自动化的仿真：开发者视角。载于 *ICST*。 [3] G. Ahuja、D. Fellars、G. Kogut、E. P. Rius、B. Sights 和 H. R. Everett。2009年。城市环境探索中自主行为的测试结果。载于 *SPIE Unmanned Syst. Technology*。 [4] M. R. Aliabadi、H. Haghighi、M. V. Asl 和 R. G. Meybodi。2020年。基于规范挖掘的信息物理系统测试预言机所面临的挑战。载于 *IKT*。 [5] E. Aljalbout、J. Xing、A. Romero、I. Akinola、C. R. Garrett、E. Heiden、A. Gupta 等。2025年。机器人领域的现实差距：挑战、解决方案与最佳实践。《控制、机器人与自主系统年度评论》（2025年）。[6] E. Allozi、A. Yilmaz、O. Ervan 和 H. Temeltas。2022年。路径规划算法的可行性分析。[7] F. Amigoni 和 V. Schiaffonati。2016年。自主机器人学中的探索性实验。载于 *MBR*。[8] A. Andrews、M. Abdelgawad 和 A. Gario。2015年。使用 CEFSM 测试自主系统的主动世界模型。载于 *MoDeVVa@MoDELS*。[9] D. Araiza-Illan、A. G. Pipe 和 K. Eder。2016年。基于智能体的刺激，用于测试人机交互中的机器人软件。载于 *MORSE*。[10] D. Araiza-Illan、D. Western、A. G. Pipe 和 K. Eder。2016年。对协作人机交互中机器人的控制代码进行系统且逼真的仿真测试。载于 *TAROS*。[11] J. F. Arango、L. M. Bergasa、P. A. Revenga、R. Barea、E. López-Guillén、C. Gómez-Huélamo、J. Araluce 和 R. Gutiérrez。2020年。基于 ROS 的自动驾驶电动汽车线控驾驶开发流程。《*Sensors-basel*》。 20，21（2020年）。

[12] H. Araujo、M. R. Mousavi 和 M. Varshosaz。2023年。机器人与自主系统的测试、验证与确认：系统综述。TOSEM 32，2（2023年）。[13] A. Babić、G. Vasiljević 和 N. Mišković。2020年。用于测试异构海洋机器人集群长期自主性的车辆在环框架。RA-L 5，3（2020年）。[14] S. Balakirsky、F. M. Proctor、C. J. Scrapper 和 T. R. Kramer。2008年。用于移动机器人软件开发的集成控制与仿真环境。载于 IDETC/CIE。[15] D. Ball、B. Upcroft、G. Wyeth、P. Corke、A. English、P. Ross、T. Patten、R. Fitch、S. Sukkarieh 和 A. Bate。2016年。农业机器人的基于视觉的障碍物检测与导航。《J. Field Robot.》33，8（2016年）。[16] D. Banerjee 和 K. Yu。2019年。使用机械臂评估基于运动的图像采集系统的集成测试自动化。IEEE Access 7（2019年）。[17] L. Baresi 和 M. Pezzè。2006年。《软件测试导论》。ENTCS 148，1（2006年）。[18] T. Barfoot、P. Furgale、B. Stenning、P. Carle、L. Thomson、G. Osinski、M. Daly 和 N. Ghafoor。2011年。火星地面冰勘探任务支持用火星车制导、导航与控制架构的现场测试。《机器人与自主系统》59，6（2011年）。[19] E. T. Barr、M. Harman、P. McMinn、M. Shahbaz 和 S. Yoo。2015年。软件测试中的预言机问题：综述。TSE 41，5（2015年）。[20] S. K. Bashetty、H. Ben Amor 和 G. Fainekos。2020年。DeepCrashTest：将行车记录仪视频转化为自动驾驶系统的虚拟碰撞测试。载于 ICRA。[21] G. Bath 和 J. McKay。2014年。《软件测试工程师手册：ISTQB 测试分析师与技术测试分析师高级级别证书学习指南》。Rocky Nook Inc。[22] C. Bédard、I. Lütkebohle 和 M. Dagenais。2022年。ros2_tracing：用于 ROS 2 实时跟踪的多用途低开销框架。RA-L 7，3（2022年）。[23] S. Bensalem、M. Bozga、M. Krichen 和 S. Tripakis。2005年。通过观察器自动生成测试实时应用的一致性。ENTCS 113（2005年）。[24] A. Bertolino。2007年。软件测试研究：成果、挑战与愿景。载于 FOSE。[25] A. Bertolino、P. Braione、G. D. Angelis、L. Gazzola、F. Kifetew、L. Mariani、M. Orrù、M. Pezzè、R. Pietrantuono、S. Russo 和 P. Tonella。2021年。基于现场的测试技术综述。CSUR 54，5（2021年）。[26] Z. Bi、Z. Miao、B. Zhang 和 C. W. Zhang。2020年。集成机器人系统测试标准的最新进展。《Rob. Comput. Integr. Manuf.》63（2020年）。[27] G. Biggs、T. Sakamoto、K. Fujiwara 和 K. Anada。2013年。在安全机器人技术中采用以模型为中心的设计方法和工具的经验。载于 IROS。[28] A. Bihlmaier 和 H. Wörn。2014年。机器人单元测试。载于 SIMPAR。[29] S. Birrell、J. Hughes、J. Y. Cai 和 F. Iida。2020年。经现场测试的卷心莴苣机器人收割系统。JFR 37，2（2020年）。[30] J. Borenstein 和 L. Feng。1995年。UMBmark：用于测量移动机器人里程计误差的基准测试。载于 SPIE Mobile Robots。[31] R. Bostelman、R. Norcross、J. Falco 和 J. Marvel。2013年。用于开发在人类附近作业的无人及有人驾驶工业车辆标准测试方法。载于《多传感器多源信息融合：架构、算法与应用》。[32] R. Bostelman、W. Shackleford、G. Cheok 和 R. Norcross。2012年。自动导引车安全标准的标准测试程序与度量指标开发。载于 PerMIS。[33] D. Bozhinoski、D. Di Ruscio、I. Malavolta、P. Pelliccione 和 I. Crnkovic。2019年。移动机器人系统的安全性：从软件工程视角开展的系统映射研究。JSS 151（2019年）。[34] G. Breitenhuber。2020年。迈向 ROS 网络的应用层测试。载于 IRC。[35] M. A. S. Brito、S. R. S. Souza 和 P. S. L. Souza。2022年。机器人系统的集成测试。《Softw. Quality J.》30，1（2022年）。[36] A. Broggi、M. Buzzoni、S. Debattisti、P. Grisleri、M. C. Laghi、P. Medici 和 P. Versari。2013年。自动驾驶技术的大规模测试。T-ITS 14，3（2013年）。[37] D. Brugali。2007年。实验机器人学的软件工程。第30卷。Springer。[38] D. Brugali。2014年。发挥机器人学与软件工程之间的协同效应：项目式实验室。载于 EDUCON。[39] D. Brugali 和 P. Salvaneschi。2006年。机器人软件开发中的稳定方面。IJARS 3，1（2006年）。[40] R. Caldas、J. A. P. García、M. Schiopu、P. Pelliccione、G. Rodrigues 和 T. Berger。2024年。基于 ROS 的机器人系统的运行时验证与现场测试。TSE 50，10（2024年）。[41] R. C. Cardoso、L. A. Dennis、M. Farrell、M. Fisher 和 M. Luckcuck。2020年。迈向模块化机器人系统的组合式验证。EPTCS 329（2020年）。

[42] L. Carlone 和 B. Bona。2009年。关于鲁棒定位的比较研究：基于距离的定位中概率滤波器的容错与鲁棒性测试。载于 ICAR。 [43] N. Casiddu、F. Burlando 和 A. Vacanti。2020年。人形机器人：可用性指南。载于 IHSED。 [44] S. Chainarong 和 K. Yaovaja。2022年。自主物料搬运车辆避障系统的开发。载于 ICT&KE。 [45] P. Cheng。2023年。基于 PID算法的水下机器人目标抓取仿真与验证。载于 INCET。 [46] A. R. Chowdhury、G. Soh、S. Foong 和 K. Wood。2018年。滚动和旋转机器人在室外地面上鲁棒路径跟踪控制的实验。《机器人与自主系统》106 (2018)。 [47] Y. Chung 和 Y.-P. Yang。2021年。基于动态路径规划和模型预测控制的自动驾驶电动汽车硬件在环仿真。《电子学》10, 19 (2021)。 [48] M. Collet、A. Gotlieb、N. Lazaar 和 M. Mossige。2019年。通过基于约束的连续轨迹生成对单臂机器人进行压力测试。载于 AITest。 [49] G. Conte、D. Scaradozzi、D. Mannocchi、P. Raspa、L. Panebianco 和 L. Screpanti。2018年。用于自主水面艇的 ROS多智能体结构的开发与实验测试。JINT 92, 3 (2018)。 [50] S. Cossell。2012年。利用企业配置测试软件，在多智能体机器人平台上实现自动化系统工程的新方法。ACRA。 [51] P. Ćwiąkała。2019年。无人飞行器（UAV）自动任务轨迹的测试流程。《应用科学》9, 17 (2019)。[52] R. Delgado、M. Campusano 和 A. Bergel。2021年。基于行为的机器人技术中的模糊测试。载于 ICRA。 [53] X. Deng、B. Zhou、X. Sun、H. Yang 和 L. Chen。2021年。一种基于定时自动机的自动导引车可靠性检测方法。《系统科学与控制工程》9, 1 (2021)。 [54] E. Dey、J. Hossain、N. Roy 和 C. Busart。2022年。SynchroSim：用于异构多机器人系统的集成协同仿真中间件。载于 DCOSS。 [55] M. G. dos Santos、S. Hallé、F. Petrillo 和 Y.-G. Guéhéneuc。2024年。AAT4IRS：工业机器人系统的自动化验收测试。《机器人学与人工智能前沿》11 (2024)。 [56] M. G. dos Santos 和 F. Petrillo。2021年。机器人系统软件工程：系统映射研究。arXiv 2102.12520 (2021)。 [57] J. A. Douthwaite、B. Lesage、M. Gleirscher、R. Calinescu、J. M. Aitken、R. Alexander 和 J. Law。2021年。用于协作机器人安全保障的模块化数字孪生框架。《机器人学与人工智能前沿》8 (2021)。 [58] P. Eckert 和 A. J. Ijspeert。2019年。多足地面机器人的敏捷性基准测试。《IEEE机器人学汇刊》35, 2 (2019)。[59] D. Erdos 和 S. E. Watkins。2008年。无人机自动驾驶仪集成与测试。载于 IEEE第5区会议。 [60] F. Erich、A. Saksena、G. Biggs 和 N. Ando。2019年。用于机器人机械臂的物理集成测试框架的设计与开发。载于系统集成国际研讨会。 [61] J. Ernits 、 E. Halling 、 G. Kanter 和 J. Vain。2015 年。ROS 软件包的基于模型的集成测试：移动机器人案例研究。载于 ECMR。 [62] V. Estivill-Castro、R. Hexel 和 C. Lusty。2018年。在去除图形用户界面的仿真中持续集成测试完整的机器人行为。载于 MODELS。 [63] J. Falco、D. Hemphill、K. Kimble、E. Messina、A. Norton、R. Ropelato 和 H. Yanco。2020年。用于评估机器人末端执行器抓取力、抓取周期时间、手指力量和手指重复性的基准测试规程。《IEEE机器人与自动化快报》5, 2 (2020)。 [64] M. J. Fernandez、P. J. Sanchez-Cuevas、G. Heredia 和 A. Ollero。2019年。使用 ROS和定制的基于ECIES的方法保障无人机通信安全。载于 RED UAS。[65] G. Fink 和 M. Bishop。1997年。基于属性的测试：一种用于保障测试的新方法。《软件工程札记》22, 4 (1997)。[66] W. Fink 和 M. A. Tarbell。2009年。用于遥控和自主监视、侦察与探索的多漫游车测试平台。载于 SPIE太空探索技术会议。 [67] L. Flückiger 和 H. Utz。2014年。空间机器人技术的面向服务的机器人架构：设计、测试与经验教训。J. Field Robot. 31, 1 (2014)。 [68] A. Fornasier、M. Scheiber、A. Hardt-Stremayr、R. Jung 和 S. Weiss。2021年。VINSEval：用于统一测试视觉惯性导航系统算法一致性与鲁棒性的评估框架。载于 ICRA。 [69] M. Gadaleta、G. Berselli、M. Pellicciari 和 F. Grassia。2021年。针对高负载工业机器人能源消耗优化的广泛实验研究及开放研究数据集。Rob. Comput. Integr. Manuf. 68 (2021)。 [70] R. Garcia 和 K. P. Valavanis。2009年。自主直升机测试平台的实现。《智能机器人系统杂志》54, 1 (2009)。

[71] S. Garcia、C. Menghi、P. Pelliccione、T. Berger 和 R. Wohlrab。2018年。一种面向分布式、协作式和自主机器人的架构。载于 ICSA。[72] S. García、D. Strüber、D. Brugali、T. Berger 和 P. Pelliccione。2020年。机器人软件工程：服务机器人领域的视角。载于 FSE。[73] S. Garcia、D. Strueber、D. Brugali、A. D. Fava、P. Pelliccione 和 T. Berger。2023年。服务机器人中的软件可变性。EMSE 28, 1 (2023)。[74] S. Garcia、D. Strueber、D. Brugali、A. D. Fava、P. Schillinger、P. Pelliccione 和 T. Berger。2019年。服务机器人的可变性建模：经验与挑战。载于 VaMoS。 [75] V. Garousi、M. Felderer、C. M. Karapıçak 和 U. Yılmaz。2018年。嵌入式软件测试：文献综述。IST 104 (2018)。 [76] N. Gasteiger、H. S. Ahn、C. Lee、J. Lim、B. A. MacDonald、G. H. Kim 和 E. Broadbent。2022年。与老年人共同参与辅助健康机器人的设计、开发与测试：一项为期四年的国际项目。《人机交互杂志》11, 4 (2022)。[77] D. I. Gertman、C. McFarland、T. A. Klein、A. E. Gertman 和 D. J. Bruemmer。2007年。一种测试无人车辆行为与自主性的方法论。载于 PerMIS。 [78] M. Gleirscher、R. Calinescu、J. Douthwaite、B. Lesage、C. Paterson、J. Aitken、R. Alexander 和 J. Law。2022年。人机协作中最优安全控制器的经验证综合。Sci. of Comp. Programming 218 (2022)。[79] M. Glinz。2007年。论非功能需求。载于 RE。[80] D. Golchinfar、D. Vaziri、D. Hennekeuser、G. Stevens 和 D. Schreiber。2023年。让我成为您的服务机器人：探索服务领域人机协作的早期用户体验。载于 RO-MAN。[81] A. Gotlieb、D. Marijan 和 H. Spieker。2020年。工业机器人系统测试：一个新的战场！载于《机器人软件工程》。[82] J. M. Gregory、D. Sahu、E. Lancaster、F. Sanchez、T. Rocks、B. Kaukeinen、J. Fink 和 S. K. Gupta。2022年。野外机器人测试与评估中的主动学习：自主越野导航案例研究。载于 ICRA。[83] J. Grzywna、A. Jain、J. Plew 和 M. Nechyba。2005年。通过虚拟飞行试验台快速开发微型飞行器的基于视觉的控制。载于 ICRA。[84] N. Gucunski、B. Basily、J. Kim、J. Yi、T. Duong、K. Dinh、S.-H. Kee 和 A. Maher。2017年。RABIT：用于改善桥面板管理的实现、性能验证及与其他机器人平台的集成。IJIRA 1, 3 (2017)。[85] Y. Guo、J. Guo、C. Liu、H. Xiong、L. Chai 和 D. He。2020年。停机坪上农业无人机的精准着陆测试与仿真。《传感器》20, 12 (2020)。[86] H. Hamann 和 A. Reina。2022年。计算与机器人学中的可扩展性。《IEEE计算机汇刊》71, 6 (2022)。[87] J. Harbin、S. Gerasimou、N. Matragkas、A. Zolotas 和 R. Calinescu。2021年。多机器人系统的模型驱动仿真分析。载于 MODELS。[88] A. M. Hass。2014年。《高级软件测试指南》。Artech House, Inc.。[89] A. Hereau、K. Godary-Dejean、J. Guiochet 和 D. Crestani。2021年。一种基于故障树的容错控制架构，用于执行横断面任务的水下机器人。载于 ICRA。[90] A. Hietanen、J. Latokartano、A. Foi、R. Pieters、V. Kyrki、M. Lanz 和 J.-K. Kämäräinen。2021年。机器人操作的位姿估计基准测试。Robot. & Auton. Syst. 143 (2021)。[91] D. Honfi、G. Molnár、Z. Micskei 和 I. Majzik。2017年。自主机器人的基于模型的回归测试。载于 SDL。[92] T. P. Huck、C. Ledermann 和 T. Kröger。2020年。用于机器人系统早期安全验证的基于仿真的测试。载于 SPCE。[93] T. P. Huck、C. Ledermann 和 T. Kröger。2022年。通过在仿真中创建危险的人类工人行为来测试机器人系统安全性。《IEEE机器人与自动化快报》7, 2 (2022)。[94] D. Humeniuk、F. Khomh 和 G. Antoniol。2022年。一种基于搜索的框架，用于自动生成信息物理系统的测试环境。IST 149 (2022)。 [95] D. Humeniuk、F. Khomh 和 G. Antoniol。2023年。Ambiegen：用于自主系统测试的基于搜索的框架。Sci. of Comp. Programming 230 (2023)。[96] C. M. Humphrey、C. Henk、G. Sewell、B. W. Williams 和 J. A. Adams。2007年。评估多机器人接口的可扩展性。载于 HRI。[97] IEEE。1990年。《软件工程术语标准词汇表》。国际标准 IEEE Std 610.12-1990。[98] IEEE/ISO/IEC。2021年。《软件与系统工程——软件测试——第4部分：测试技术》。国际标准 ISO/IEC/IEEE 29119-4。[99] C. Insam、A. Kist 和 D. J. Rixen。2020年。采用迭代学习控制的高保真实时混合子结构测试。载于 ISR。[100] 国际软件测试资格委员会（ISQTB）。[未注明日期。]。《软件测试术语标准词汇表》。http://glossary.istqb.org/。

[101] 国际软件测试资格委员会（ITSQB）。2024年。《认证测试员——基础级大纲》。[102] ISO/IEC/IEEE。2013年。《软件与系统工程——软件测试——第3部分：测试文档》。国际标准 ISO/IEC/IEEE 29119-3。[103] ISO/IEC/IEEE。2013年。《软件与系统工程——软件测试——第1部分：概念与定义》。国际标准 ISO/IEC/IEEE 29119-1。[104] ISO/IEC/IEEE。2013年。《软件与系统工程——软件测试——第2部分：测试过程》。国际标准 ISO/IEC/IEEE 29119-2。[105] A. Jacoff、H.-M. Huang、E. Messina、A. Virts 和 A. Downs。2010年。用于移动机器人性能评估的综合标准测试套件。载于 PerMIS。[106] A. Jacoff、E. Messina 和 J. Evans。2001年。自主移动机器人的参考测试路线。载于 SPIE无人地面车辆技术会议。[107] Z. Jiang、B. Chen、Z. Ju、Y. Li、Y. Xu 和 Y. Zhao。2023年。用于火力发电厂水墙检测的爬壁机器人设计与分析。《JFR》（2023年）。[108] A. Jiménez-González、J. R. Martinez-de Dios 和 A. Ollero。2013年。泛在机器人学的测试平台：综述。《Robot. & Auton. Syst.》61卷第12期（2013年）。[109] A. Jiménez-Ramírez、J. Chacón-Montero、T. Wojdynsky 和 J. González Enríquez。2023年。机器人流程自动化项目中的自动化测试。《软件：演化与过程》35卷第3期（2023年）。[110] G. Jochmann、F. Blümel、O. Stern 和 J. Roßmann。2014年。虚拟空间机器人测试平台：用于开发和评估机器人探测任务组件的综合手段。《KI》28卷第2期（2014年）。[111] M. R. Kamandar、J. Massah 和 M. Jamzad。2022年。绿篱修剪机器人的设计与评估。《计算机与电子农业》199（2022年）。[112] Y. Kaneshige、S. Satake、T. Kanda 和 M. Imai。2021年。如何克服移动社交机器人的编程与调试难题？载于人机交互。[113] J. S. Kang 和 H. S. Park。2012年。基于网络的基于组件的机器人软件自动化黑盒测试框架。载于泛在计算。[114] J. S. Kang 和 H. S. Park。2015年。用于基于组件的软件系统测试的预期结果自动生成算法。《IST》57（2015年）。[115] S. Kang、H. Hao、Q. Dong、L. Meng、Y. Xue 和 Y. Wu。2022年。面向自动驾驶仿真的基于行为树的场景规范与测试用例生成。载于 ICITES。[116] G. Kanter 和 J. Vain。2019年。TestIt：面向 ROS 的开源、可扩展的长期自主性测试工具包。载于 DESSERT。[117] G. Kanter 和 J. Vain。2020年。使用 TestIt 对自主机器人进行基于模型的测试。《可靠性与智能环境》6卷第1期（2020年）。[118] N. Kapetanović、J. Goričanec、I. Vatavuk、I. Hrabar、D. Stuhne、G. Vasiljević、Z. Kovačić、N. Mišković、N. Antolović、M. Anić 和 B. Kozina。2022年。葡萄种植与海水养殖中的异构自主机器人系统：车辆开发与系统集成。《Sensors》22卷第8期（2022年）。[119] D. S. Katz、C. Hutchison、M. Zizyte 和 C. L. Goues。2020年。将执行异常检测用作自主软件鲁棒性的判定预言机。载于 ICRA。[120] P. Kaur、Z. Liu 和 W. Shi。2022年。移动社交机器人的模拟器：最新进展与挑战。载于 MetroCAD。[121] M. E. Khan 和 F. Khan。2012年。白盒、黑盒和灰盒测试技术的比较研究。《国际先进计算机科学与应用杂志》3卷第6期（2012年）。[122] H. Khosrowjerdi 和 K. Meinke。2018年。利用时空需求对自主系统进行基于学习的测试。载于 MASES。[123] J. Kim、J. M. Esposito 和 V. Kumar。2005年。一种用于测试和验证多机器人控制器的基于快速探索随机树的算法。技术报告。[124] S. Kim 和 T. Kim。2022年。RoboFuzz：在机器人操作系统（ROS）上对机器人系统进行模糊测试，以发现正确性缺陷。载于 ESEC/FSE。[125] S. Kim、O. A. Philip、A. Tullu 和 S. Jung。2023年。面向无人飞行器的基于 ROS 的多自由度飞行测试系统的开发与验证。《IEEE Access》11（2023年）。[126] T. Kim、C. H. Kim、J. Rhee、F. Fei、Z. Tu、G. Walkup、X. Zhang、X. Deng 和 D. Xu。2019年。RVFUZZER：通过控制引导测试发现机器人车辆中的输入验证缺陷。载于 USENIX安全研讨会。[127] K. Kimble、K. Van Wyk、J. Falco、E. Messina、Y. Sun、M. Shibata、W. Uemura 和 Y. Yokokohji。2020年。评估小型零件机器人装配系统的基准测试规程。《IEEE机器人与自动化快报》5卷第2期（2020年）。[128] F. Klück、Y. Li、J. Tao 和 F. Wotawa。2023年。在自动化与自主驾驶系统领域对组合测试与基于搜索的测试进行实证比较。《IST》160（2023年）。[129] M. H. Korayem、K. Khoshhal 和 H. Aliakbarpour。2005年。用于机器人性能测试的基于视觉的仿真与实验。《国际先进制造技术杂志》25卷第11期（2005年）。

[130] J. A. Kramer 和 R. R. Murphy。2006年。无人地面车辆验收测试。载于 SPIE无人系统技术会议。[131] V. Krueger、F. Rovida、B. Grossmann、R. Petrick、M. Crosby、A. Charzoule、G. Martin Garcia、S. Behnke、C. Toscano 和 G. Veiga。2019年。测试认知机器人在制造业中的垂直与信息物理集成。《Robotics and Computer-Integrated Manufacturing》57（2019年）。[132] D. Kuehn、M. Schilling、T. Stark、M. Zenzes 和 F. Kirchner。2017年。类人机器人Charlie的系统设计与测试。《JFR》34卷第4期（2017年）。[133] M. Lamboley、C. Proy、L. Rastel、T. N. Trong、A. Zashchirinski 和 S. Buslaiev。1995年。Marsokhod：在类火星地形上的自主导航测试。《Auton. Robots》2卷第4期（1995年）。[134] B. Lee、J. Brookshire、R. Yahata 和 S. Samarasekera。2022年。迈向具备人机交互能力的安全、逼真机器人系统测试平台。载于 ICRA。[135] N. E. Leonard、D. A. Paley、R. E. Davis、D. M. Fratantoni、F. Lekien 和 F. Zhang。2010年。蒙特雷湾自适应海洋采样实地实验中的水下滑翔机编队协同控制。《JFR》27卷第6期（2010年）。[136] L. Lestingi、M. Askarpour、M. M. Bersani 和 M. Rossi。2021年。形式化验证的人机交互部署框架。《IEEE Access》9（2021年）。[137] G.-Y. Li、R.-T. Soong、J.-S. Liu 和 Y.-T. Huang。2019年。用于自主建筑巡检任务的无人机实时感知与飞行任务控制系统集成。载于 TAAI。[138] A. Liaqat、W. Hutabarat、D. Tiwari、L. Tinkler、D. Harra、B. Morgan、A. Taylor、T. Lu 和 A. Tiwari。2019年。制造业中的自主移动机器人：交通规则的制定、仿真与测试。《国际先进制造技术杂志》104卷第9期（2019年）。[139] R. Lill 和 F. Saglietti。2014年。自主机器人智能体协作的测试。载于 ICSOFT-EA。[140] J.-H. Lim、S.-H. Song、J.-R. Son、T.-Y. Kuc、H.-S. Park 和 H.-S. Kim。2010年。一种机器人平台及其组件的自动化测试方法。《JSEA》4卷第3期（2010年）。[141] Z. Liu、W. Liu、Y. Qin、F. Xiang、M. Gou、S. Xin、M. A. Roa、B. Calli、H. Su、Y. Sun 和 P. Tan。2022年。OCRTOC：面向机器人抓取与操作的云端竞赛与基准测试。《IEEE机器人与自动化快报》7卷第1期（2022年）。[142] M. Lochau、S. Peldszus、M. Kowal 和 I. Schaefer。2014年。基于模型的测试。载于 SFM。[143] R. Looije、M. A. Neerincx 和 K. V. Hindriks。2017年。阐明并测试社交机器人的设计原理，以促进儿童行为改变。《Cogn. Syst. Res.》43（2017年）。[144] M. Luckcuck、M. Farrell、L. Dennis、C. Dixon 和 M. Fisher。2020年。自主机器人系统的形式化规约与验证：综述。《CSUR》52卷第5期（2020年），41页。[145] C. L. Lundberg、H. E. Sevil、D. Behan 和 D. O. Popa。2022年。通过患者陪护和患者助行任务开展护理辅助机器人应用与人体受试测试。《Robotics》11卷第3期（2022年）。[146] M. Luperto、M. Romeo、J. Monroy、A. Vuono、N. Basilico、J. Gonzalez-Jimenez 和 N. A. Borghese。2021年。我的机器人在做什么？通过远程监督支持老年人独立生活的机器人：一项实地研究。载于 ECMR。[147] S. Macenski、S. Singh、F. Martín 和 J. Ginés Clavero。2023年。用于机器人路径跟踪的受限纯跟踪。《Auton. Robot.》47卷第6期（2023年）。[148] Ş. Maftei 和 D. Novischi。2020年。用于评估自组织机器人群的多机器人二维仿真器。载于 AQTR。[149] A. Mannion、S. Summerville、E. Barrett、M. Burke、A. Santorelli、C. Kruschke、H. Felzmann、T. Kovacic、K. Murphy、D. Casey 等。2020年。在长期住宿照护机构中向痴呆症患者介绍社交机器人MARIO：反思。《Int. J. Soc. Robot.》12卷第2期（2020年）。[150] E. P. Marcosig、J. I. Giribet 和 R. Castro。2018年。DEVS-over-ROS（Dover）：一种基于模型连续性的机器人系统仿真驱动嵌入式控制框架。载于 Winter Simulation Conference。[151] J. Marquez、C. Sullivan、R. M. Price 和 R. C. Roberts。2023年。采用行动者-评论家深度强化学习算法的硬件在环软体机器人测试框架。《IEEE机器人与自动化快报》8卷第9期（2023年）。[152] A. Marrella、L. Wang、L. Iocchi 和 D. Nardi。2022年。机器人竞赛中人机交互团队协作任务的设计与评估方法论。《J. Hum.-Robot Interact.》11卷第3期（2022年），22页。[153] E. Martin-Martin、M. Montenegro、A. Riesco、J. Rodríguez-Hortalá 和 R. Rubio。2023年。使用可执行规约语言验证 ROS NavFn规划器。132（2023年）。[154] Y. Maruyama、S. Kato 和 T. Azumi。2016年。探索 ROS 2 的性能。载于 EMSOFT。[155] T. Mashrur、Z. Ghulam、G. French 和 H. A. Abdullah。2023年。上肢障碍人士辅助进食机器人——测试与验证。《IJARS》20卷第4期（2023年）。[156] F. Mauch、A. Roennau、G. Heppner、T. Buettner 和 R. Dillmann。2017年。实地服务机器人：BratWurst Bot。载于 ICAR。[157] V. Mayoral-Vilches、R. White、G. Caiazza 和 M. Arguedas。2022年。SROS2：适用于 ROS 2 的实用网络安全工具。载于 IROS。[158] S. McConnell。1998年。《软件项目生存指南》。培生教育。

[159] R. K. Megalingam、A. Rajendraprasad 和 S. K. Manoharan。2020年。使用 ROS 导航栈比较规划路径与实际行驶路径。载于 INCET。[160] M. Mercier、S. Phillips、M. Shubert 和 W. Dong。2020年。多智能体相对制导、导航与控制算法的地面测试。载于 PLANS。[161] G. Metta、P. Fitzpatrick 和 L. Natale。2006年。YARP：又一个机器人平台。《IJARS》3，1（2006年）。[162] N. Michael、D. Mellinger、Q. Lindsey 和 V. Kumar。2010年。GRASP多微型无人机测试台。《RAM》17，3（2010年）。[163] Z. Micskei、Z. Szatmári、J. Oláh 和 I. Majzik。2012年。自主系统上下文感知行为鲁棒性与安全性的测试概念。载于 KES-AMSTA。[164] R. Mihali、M. Grigorian 和 T. Sobh。2000年。一级方程式赛车换胎机器人的优化与测试。《智能机器人系统杂志》29，3（2000年）。[165] B. P. Miller、L. Fredriksen 和 B. So。1990 33，12 1990 。UNIX实用程序可靠性的实证研究。《Commun. ACM》（ ）。[166] J. F. Montgomery、A. E. Johnson、S. I. Roumeliotis 和 L. H. Matthies。2006年。喷气推进实验室自主直升机测试台：行星探测技术研究与开发平台。《JFR》23，3-4（2006年）。[167] M. K. Moon 和 S.-C. Kim。2017年。面向老年人的移动辅助服务机器人可用性评估。载于 AHFE。[168] A. J. Moshayedi、J. Li 和 L. Liao。2021年。自动导引车（AGV）的仿真研究与 PID 调参。载于 CIVEMSA。[169] M. Mossige、A. Gotlieb 和 H. Meling。2015年。使用约束编程和持续集成测试机器人控制器。《IST》57（2015年）。[170] G. E. Mullins、A. G. Dress、P. G. Stankiewicz、J. D. Appler 和 S. K. Gupta。2018年。通过模仿学习加速自动驾驶车辆的测试与评估。载于 ICRA。[171] A. Munawar、Y. Wang、R. Gondokaryono 和 G. S. Fischer。2019年。用于仿真复杂机器人与环境的实时动态模拟器及配套前端表示格式。载于 IROS。[172] P. Muñoz、A. Cesta、A. Orlandini 和 M. D. R-Moreno。2014年。地面自主性测试环境的初步探索。《SMC-IT/SCC》。[173] V. Nagrath 和 C. Schlegel。2021年。SmartTS：机器人软件生态系统中软件测试的基于组件与模型驱动方法。《国际先进计算机科学与应用杂志》12，7（2021年）。[174] F. Neri、M. Forlini、C. Scoccia、G. Palmieri 和 M. Callegari。2023年。协作机器人避碰技术的实验评估。《Appl. Sci.》13，5（2023年）。[175] M. Nguyen、S. Wrede 和 N. Hochgeschwender。2025年。机器人系统的自动化行为驱动验收测试。载于 IROS。[176] W. R. Norris 和 A. E. Patterson。2019年。现场机器人的系统级测试与评估计划：附测试课程布局的教程。《Robotics》8，4（2019年）。[177] A. Norton、P. Gavriel 和 H. Yanco。2019年。评估自动导引车和自主移动机器人导航与避障能力的标准测试方法。《SSMS》3，2（2019年）。[178] M. Núñez、R. M. Hierons 和 R. Lefticaru。2023年。循环系统的实现关系与测试：引入概率。《Robot. & Auton. Syst.》165（2023年）。[179] M. Nybacka。2009年。汽车冬季测试的机遇。[180] M. O. Okwu、L. K. Tartibu、D. R. Enarevba、O. J. Oyejide、O. B. Otanocha 和 S. Adumene。2022年。轻型自主割草机的开发及其基于模糊逻辑技术的性能分析。载于 icABCD。[181] A. Ortega、N. Hochgeschwender 和 T. Berger。2022年。服务机器人现场测试：经验报告。载于 IROS。[182] A. Ortega、S. Parra、S. Schneider 和 N. Hochgeschwender。2024年。用于移动机器人仿真测试的可组合且可执行的场景。《机器人学与人工智能前沿》11（2024年）。[183] D. Paez-Granados、Y. He、D. Gonon、D. Jia、B. Leibe、K. Suzuki 和 A. Billard。2022年。自主群体导航中的行人与机器人交互：反应式控制方法与评估指标。载于 IROS。[184] A. Paikan、S. Traversaro、F. Nori 和 L. Natale。2015年。用于机器人系统测试驱动开发的通用测试框架。载于 MESAS。[185] J. Pak、J. Kim、Y. Park 和 H. I. Son。2022年。智慧农场中自主移动机器人路径规划算法的现场评估。《IEEE Access》10（2022年）。[186] L. Palermino、A. M. M. Fathy、M. Carnevale 和 H. Giberti。2022年。面向工业应用的避碰算法开发与测试。[187] Q. Pang、Y. Yuan 和 S. Wang。2022年。MDPFuzz：测试求解马尔可夫决策过程的模型。载于 ISSTA。[188] J. Park、M. Kang、Y. Lee、J. Jung、H.-T. Choi 和 J. Choi。2023年。用于海洋环境中自主协同导航任务的多艘自主水面艇：开发与初步现场测试。《IEEE Access》11（2023年）。

[189] S. Parra、A. Ortega、S. Schneider 和 N. Hochgeschwender。2023年。一千个世界：用于移动机器人导航栈仿真测试的场景规格说明与生成。载于 IROS。[190] G. Paul、S. Webb、D. Liu 和 G. Dissanayake。2010年。用于钢桥维护的机器人系统：现场测试。载于 ACRA。[191] S. Peldszus、D. Brugali、D. Strueber、P. Pelliccione 和 T. Berger。2025年。机器人学中的软件重配置。《EMSE》30，3（2025年）。[192] É. Piel、A. González 和 H.-G. Gross。2010年。大型发布/订阅系统集成测试的自动化。载于《分布式基于事件的系统的原理与应用》。[193] N. Pinrath 和 N. Matsuhira。2020年。将虚拟机器人环境平台与机器人操作系统集成，实现机器人的实时仿真。载于 SICE。[194] J. Platt 和 K. Ricks。2022年。面向移动地面机器人仿真的 ros-unity3d 与 ros-gazebo 比较分析。《智能机器人系统杂志》106，4（2022年）。[195] J. Poncela 和 M. Aguayo-Torres。2013年。无线水下机器人测试框架。《Wireless Personal Communications》70，3（2013年）。[196] P. K. Pook 和 C. K. DeBolt。1999年。用于协同子弹药清除的试验台机器人开发。《IJRR》18，7（1999年）。[197] D. Powell、J. Arlat、H. N. Chu、F. Ingrand 和 M.-O. Killijian。2012年。测试自主系统实时控制软件的输入时序鲁棒性。载于 EDCC。[198] M. Proetzsch、F. Zimmermann、R. Eschbach、J. Kloos 和 K. Berns。2010年。使用领域特定语言对自主移动机器人进行系统化测试的方法。载于 KI。[199] G. Punzo、C. MacLeod、K. Baumanis、R. Summan、G. Dobie、G. Pierce 和 M. Macdonald。2019年。用于满足安全约束的自主空中巡检的二分式制导、导航与控制架构。《智能机器人系统杂志》95，3（2019年）。[200] R. Queiroz、T. Berger 和 K. Czarnecki。2019 年。GeoScenario：一种用于自动驾驶场景表示的开放式领域专用语言。载于 IV。[201] R. Queiroz、D. Sharma、R. Caldas、K. Czarnecki、S. García、T. Berger 和 P. Pelliccione。2024年。用于自动驾驶系统基于场景测试的驾驶员-车辆模型。《T-ITS》25，8（2024年）。[202] M. Quigley、B. Gerkey、K. Conley、J. Faust、T. Foote、J. Leibs、E. Berger、R. Wheeler 和 A. Ng。2009年。ROS：一种开源机器人操作系统。《ICRA Workshop Open Source Softw.》3，3.2（2009年）。[203] I. Reithner、M. Papa、M. Aburaia、W. Wöber 和 C. Ambros。2021年。通过移动机器人和生产网络实例展示安全性与信息安全之间交互的分析。载于 DAAAM Int. Symp.。[204] C. Robert、J. Guiochet 和 H. Waeselynck。2020年。在仿真中测试非确定性机器人——需要重复运行多少次？载于 IRC。[205] C. Robert、T. Sotiropoulos、H. Waeselynck、J. Guiochet 和 S. Vernhes。2020年。《绿野仙踪》：在仿真中测试农业机器人。《EMSE》25，3（2020年）。[206] A. Rodic、M. Jovanovic、S. Popic 和 G. Mester。2011年。面向信息结构化环境中网络化机器人系统研究、开发与测试的可扩展实验平台：无线机器人传感器网络实验测试平台。[207] ROS 2文档。 [无日期。]。测试。访问日期：2025-06-18。[208] ROS Wiki。 [无日期。]。使用 ROS 进行自动化测试。访问日期：2025-06-18。[209] M. D. Rosa、S. Goldstein、P. Lee、J. Campbell 和 P. Pillai。2008年。分布式观察点：大型模块化机器人系统的调试。《IJRR》27，3-4（2008年）。[210] N. B. Ruparelia。2010年。软件开发生命周期模型。《软件工程札记》35，3（2010年）。[211] A. Sadek、G. Bono、B. Chidlovskii 和 C. Wolf。2022年。对机器人在真实环境中导航时传感器使用与视觉推理的深入实验研究。载于 ICRA。[212] O. Sadka、J. Giron、D. Friedman、O. Zuckerman 和 H. Erel。2020年。虚拟现实作为非人形社交机器人的仿真工具。载于 CHI。[213] F. Saglietti 和 M. Meitner。2016年。机器人协作与重配置的模型驱动结构化及统计测试。载于 MORSE。[214] A. Santos、A. Cunha 和 N. Macedo。2018年。机器人操作系统的基于属性的测试。载于 A-TEST。[215] A. Santos、A. Cunha 和 N. Macedo。2021年。高保障ROS框架。载于 RoSE。[216] A. Santos、A. Cunha 和 N. Macedo。2022年。面向消息的系统的模式引导测试。《ENASE》。[217] M. G. D. Santos、F. Petrillo、S. Hallé 和 Y.-G. Guéhéneuc。2022年。一种将自动化验收测试应用于工业机器人系统的方法。载于 IRC。[218] A. Sarkar 和 K. Czamecki。2019年。一种行为驱动方法，用于对自动驾驶车辆的稀有事件情形进行采样。载于 IROS。

[219] D. Sartori、F. Quagliotti、M. J. Rutherford 和 K. P. Valavanis。2014年。固定翼无人机反步控制器自动驾驶仪的实现与测试。76，3（2014年）。[220] L. V. Sartori、J. Guiochet、H. Waeselynck、A. A. B. Galvan、S. Hébert-Vernhes 和 M. Albert。2022年。将测试生成集成到基于仿真的平台中：经验报告。载于 AST。[221] I. Satoh。2019年。开发和测试用于移动机器人的网络软件。载于 ENASE。[222] F. A. Schiegg、J. Krost、S. Jesenski 和 J. Frye。2019年。用于高级驾驶辅助系统设计与测试的新型仿真框架。载于 VTC。[223] C. Schlegel、T. Haßler、A. Lotz 和 A. Steck。2009年。机器人软件系统：从代码驱动到模型驱动设计。载于 ICAR。[224] C. Schlegel、A. Steck、D. Brugali 和 A. Knoll。2010年。机器人学中的设计抽象与流程：从代码驱动到模型驱动工程。载于 SIMPAR。[225] M. Seleck`y、J. Faigl 和 M. Rollo。2019年。混合现实仿真在多无人机系统渐进式开发中的应用分析。《智能机器人系统杂志》95，1（2019年）。[226] H. Seyyedhasani、C. Peng、W.-j. Jang 和 S. G. Vougioukas。2020年。收获期间人工采摘者与作物运输机器人协作——第二部分：模拟器评估与机器人调度案例研究。《计算机与电子农业》172（2020年）。[227] H. Seyyedhasani、C. Peng、W. jiunn Jang 和 S. G. Vougioukas。2020年。收获期间人工采摘者与作物运输机器人协作——第一部分：模型与模拟器开发。《计算机与电子农业》172（2020年）。[228] D. A. Shell 和 J. M. O'Kane。2020年。现实作为现实的仿真：机器人幻觉、根本限制及一项物理演示。载于 ICRA。[229] Y.-J. Shin、E. Cho、H. Kim 和 D.-H. Bae。2022年。多控制器 CPS 的实地操作测试数据集：自动驾驶的建模案例研究。载于 SoSE。[230] A. Sibilska-Mroziewicz、J. Możaryn、A. Hameed、M. M. Fernández 和 A. Ordys。2022年。基于仿真的控制设计评估框架：以多体机器人系统中的蛇形机器人为例。55，4（2022年）。[231] K. Sim、J. H. Yi 和 H. Cho。2023年。SMINER：基于单元测试用例检测软件系统中未受限及实现错误的行为。CMC 75，2（2023年）。[232] Ł. Sobczak、K. Filus、J. Domańska 和 A. Domański。2022年。利用 UDP 桥接技术为无人地面车辆构建实时测试平台。Sensors 22，21（2022年）。[233] S. O. Sohail、A. Mitrevski、N. Hochgeschwender 和 P. G. Ploger。2021年。基于仿真的基于属性的测试：验证机器人在桌面操作中的动作执行。载于 ECMR。[234] S. O. Sohail、S. Schneider 和 N. Hochgeschwender。2023年。对机器人进行标准符合性自动化测试。载于 CASE。[235] Q. Song、E. Engström 和 P. Runeson。2021年。自主系统测试概念：学术文献与行业实践。载于 WAIN。[236] T. Sotiropoulos、G. Guiochet、I. Ingrand 和 W. Waeselynck。2016年。用于测试机器人导航的虚拟世界：难度等级研究。载于 EDCC。[237] T. Sotiropoulos、H. Waeselynck、J. Guiochet 和 F. Ingrand。2017年。能否在仿真中发现机器人导航漏洞？一项探索性研究。载于 QRS。[238] J. Straub 和 J. Huber。2013年。使用人工智能测试两个人工智能系统的效用分析。《计算机》2，2（2013年）。[239] A. Suarez、H. Romero、R. Salmoral、J. A. Acosta、J. Zambrano 和 A. Ollero。2021年。用于安装夹式驱鸟器的空中操作机器人实验评估：室外飞行测试。载于 AIRPHARO。[240] V. SunSpiral、D. Wheeler、D. Chavez-Clemente 和 D. Mittman。2012年。ATHLETE 机器人 FootFall 规划系统的开发与现场测试。JFR 29，3（2012年）。[241] S. Swanborn 和 I. Malavolta。2021年。Robot Runner：一款用于自动执行机器人软件实验的工具。载于 ICSE Companion Proc.。[242] A. Sweet、G. Gorospe、M. Daigle、E. Balaban、I. Roychoudhury、S. Narasimhan 等。2014年。在硬件移动机器人测试平台上演示支持预测性维护的决策算法。载于 PHM Conf.。[243] M. Tanaka、K. Kon、M. Nakajima、N. Matsumoto、S. Fukumura、K. Fukui、H. Sawabe、M. Fujita 和 K. Tadakuma。2020年。用于工厂灾害防范的铰接式移动机器人 T2 Snake-4 的开发与现场测试。AR 34，2（2020年）。[244] L. Tang、E. Hettler、B. Zhang 和 J. DeCastro。2011年。用于实时自主车辆预测性维护与应急管理应用的测试平台。载于 PHM Conf.。[245] S. Tang 、 G. Liu 、 Z. Lin 、 X. Li 和 M. Pan。2020 年。集成测试平台上工业机器人伺服系统的测试流程优化方法。Sfi. S. Sci. C. 2020，1（2020年）。[246] S. Teixeira、R. Arrais 和 G. Veiga。2021年。用于机器人持续集成与部署的云仿真。载于 INDIN。

[247] C. S. Timperley、A. Afzal、D. S. Katz、J. M. Hernandez 和 C. Le Goues。2018年。撞毁模拟飞机成本低廉：仿真能否及早发现机器人漏洞？载于 ICST。[248] E. Torta、R. H. Cuijpers、J. F. Juola 和 D. Van Der Pol。2012年。人形机器人近体行为的建模与测试。9，04（2012年）。[249] N. Totev。2021年。机器人系统测试：案例研究。载于 ISGT（CEUR Workshop Proc.，第2933卷）。[250] W. Touzout、D. Benazzouz 和 Y. Benmoussa。2022年。将移动机器人能量建模集成到基于 ROS 和 Gazebo 的仿真环境中。Mechatron. Syst. Control. 50，2（2022年）。[251] T. Verdu、N. Maury、P. Narvor、F. Seguin、G. Roberts、F. Couvreux、G. Cayez、M. Bronz、G. Hattenberger 和 S. Lacroix。2020年。无人机云层探测自适应模式的实验飞行。载于 IROS。[252] B. Vieira、R. Severino、E. V. Filho、A. Koubaa 和 E. Tovar。2019年。COPADRIVe——面向协同自动驾驶应用的逼真仿真框架。载于 ICCVE。[253] N. I. Vitzilaios 和 N. C. Tsourveloudis。2008年。使用原型试验台控制小型直升机的高度。载于 ICINCO。[254] N. I. Vitzilaios 和 N. C. Tsourveloudis。2009年。小型无人直升机的实验测试平台。《智能机器人系统杂志》54，5（2009年）。[255] B. Wang、R. Wang 和 H. Song。2022年。利用差分模糊测试提升工业机器人系统的可信性。IEEE Trans. Ind. Informat. 19，3（2022年）。[256] C. Wang、Y. C. Tok、R. Poolat、S. Chattopadhyay 和 M. R. Elara。2021年。如何保障自主移动机器人的安全？采用模糊测试、检测与缓解的方法。《系统架构杂志》112（2021年）。[257] S. Wang。2020年。自主服务机器人长期部署的可靠性工程。博士论文。[258] A. Wąsowski 和 T. Berger。2023年。领域特定语言。Springer。[259] M. Watanabe、Y. Ozawa、K. Takahashi、T. Kimura、K. Tadakuma、G. Marafioti 和 S. Tadokoro。2022年。用于在碎屑锥中搜寻幸存者的两轮机器人平台的硬件设计与测试。载于 SSRR。[260] M. Webster、D. Western、D. Araiza-Illan、C. Dixon、K. Eder、M. Fisher 和 A. G. Pipe。2020年。人机团队验证与确认的协同方法。IJRR 39，1（2020年）。[261] H. Wei、J. Timmis 和 R. Alexander。2017年。演进测试环境以识别群体机器人算法中的故障。载于 CEC。[262] V. K. Weinstein 和 S. M. LaValle。2024年。机器人的等价环境与覆叠空间。载于 ESIAM。[263] B. A. Weiss 和 L. C. Schmidt。2012年。多关系评估设计：自动测试计划生成器的建模。载于 PerMIS。[264] B. Weng、G. A. Castillo、W. Zhang 和 A. Hereid。2022年。通过场景采样进行安全测试、验证与表征：足式机器人的案例研究。载于 IROS。[265] D. Wettergreen、S. Moreland、K. Skonieczny、D. Jonak、D. Kohanbash 和 J. Teza。2010年。原型月球勘探器的设计与现场试验。IJRR 29，12（2010年）。[266] J. Wienke、D. Wigand、N. Koster 和 S. Wrede。2018年。机器人软件组件的基于模型的性能测试。载于 IRC。[267] J. Wienke 和 S. Wrede。2017年。机器人系统组件的性能回归测试与运行时验证。AR 31，22（2017年）。[268] D. B. Wilson、A. Goktogan 和 S. Sukkarieh。2012年。无人机会合：从概念到飞行试验。载于 ACRA。[269] P. Wolf、T. Groll、S. Hemer 和 K. Berns。2020年。机器人模拟器的演进：利用虚幻引擎4，在非结构化环境中实现对复杂自主机器人的真实世界质量测试。载于 SIMULTECH。[270] T. Woodlief、S. Elbaum 和 K. Sullivan。2021年。对移动机器人环境进行模糊测试，以快速自动检测崩溃。载于 ICRA。[271] K.-T. Xie、J.-J. Bai、Y.-H. Zou 和 Y.-P. Wang。2022年。ROZZ：面向 ROS 机器人程序的基于属性的模糊测试。载于 ICRA。[272] Y. Xiong、Y. Ge、L. Grimstad 和 P. J. From。2020年。一种自主草莓采摘机器人：设计、开发、集成与现场评估。JFR 37，2（2020年）。[273] Y. Xiong、C. Peng、L. Grimstad、P. J. From 和 V. Isler。2019年。带有绳驱夹爪的草莓采摘机器人的开发与现场评估。《计算机与电子农业》157（2019年）。[274] T. Xu、D. Wang、Z. Xiao、C. Chu 和 W. Zhang。2021年。一种用于评估自主铰接车辆路面压实性能的四级测试系统。IJARS 18，1（2021年）。[275] S. Yoon 和 R. Bostelman。2019年。自动至自主——无人地面车辆（A-UGV）性能标准分析。载于 RoSE。[276] S. N. Yu、S. Y. Lee、C. S. Han、K. Y. Lee 和 S. H. Lee。2007年。幕墙安装机器人的开发：施工现场的性能与效率测试。《自主机器人》22，3（2007年）。[277] O. Zendel、W. Herzner 和 M. Murschitz。2013年。VITRO——基于模型的鲁棒性视觉测试。载于 ISR。

[278] J. Zenzeri、D. De Santis、V. Mohan、M. Casadio 和 P. Morasso。2013年。基于被动运动范式，使用功能性伸展测试探测人形机器人的双足站立静态稳定性。机器人学期刊 2013年，1（2013年）。[279] J. Zhang、Z. Zheng、B. Yin、K. Qiu 和 Y. Liu。2019年。使用蜕变测试对基于图搜索的路径规划算法进行测试。载于 *PRDC*。[280] H. Zhou、S. Zhang、J. Zhang、C. Zhang、S. Wang、Y. Zhai 和 W. Li。2022年。割胶机器人的设计、开发与现场评估。*JFR* 39，1（2022年）。[281] M. R. Zofka、L. Töttel、M. Zipfl、M. Heinrich、T. Fleck、P. Schulz 和 J. M. Zöllner。2020年。推动 ROS 走向黑暗面：面向自动驾驶车辆混合现实测试系统的基于 ROS 的联合仿真架构。载于 *MFI*。