

基于大疆机甲大师的信息物理综合测试床

Cyber-Physical Integrated Test Bed for DJI RoboMaster

毛心怡^{1,2}, 秦逸^{1,2}, 许畅^{1,2}

¹ (计算机科学与技术系, 南京大学)

² (计算机软件新技术国家重点实验室, 南京大学)

摘要: 信息物理融合系统软件 (简称 CPS 软件) 的开发过程通常涉及模型在环、软件在环和硬件在环三个阶段。模型/软件在环阶段通常使用模拟器/仿真器来模拟 CPS 软件和物理平台的交互过程, 而硬件在环阶段则直接使用目标物理平台完善和继续提升 CPS 软件的功能性和可靠性。然而考虑到物理平台和运行环境的复杂性, 模拟器/仿真器往往难以精确还原 CPS 软件和物理平台的交互, 特别是交互中存在的非确定性; 而使用物理平台则会带来较大的运行开发成本, 同时也较难实现对在物理平台上运行的 CPS 软件状态的实时监控和操控。本文提出并实现了一个基于大疆机甲大师的信息物理综合测试床, 用于为 CPS 软件提供高效且真实的软/硬件交互过程, 保证软件运行时的安全性、可观测性和可操作性。测试床由基于虚拟引擎的 SiL Wrapper 模块和基于物理环境的 HiL Wrapper 模块组成, 通过基于四向传感距离的定位技术和基于 API 中断的姿态调整技术, 支撑了两个模块的高效准确交互, 为 SDK 接口调用返回高质量的软硬件交互结果。

关键词 信息物理融合系统; 软件开发; 软件在环; 硬件在环; 无人小车

信息物理融合系统^[1](Cyber-Physical Systems, 简称 CPS)是由一组高度融合的软件组件和物理环境构成的工程系统。不同于传统软件, CPS 软件除了功能性任务之外, 还承担了连接信息与物理空间的交互任务, 而物理空间高度动态非确定的特点为 CPS 软件的开发带来了新的挑战。

CPS 的软件开发过程按其开发环境可以分为模型在环(MiL)、软件在环(SiL)和硬件在环(HiL)三个阶段^[2], 其中模型/软件在环和外部环境交互时使用模拟器来降低运行场景搭建的代价, 由于模拟器和真实硬件差异较大, 仅使用虚拟环境进行开发无法保证软件的可靠性; 硬件在环阶段则保证了软件和真实硬件的交互, 然而实现硬件在环开发往往需承担更大的搭建成本和调试风险^[2]。

已有一些工作尝试结合二者优势设计介于软件在环和硬件在环之间的综合开发框架^[2-3]。但这些框架未解决开发环境搭建代价高, 虚拟仿真环境受限于物理环境, 程序运行时可观测性、可操作性不足等问题。

本文搭建了基于大疆机甲大师的信息物理综合测试床, 通过对机甲大师软件开发工具包的扩展, 为大疆机甲大师这类 CPS 提供高效且真实的软/硬件交互过程。该平台有效结合了 CPS 软件开发中的软件在环和硬件在环阶段, 通过搭建基于虚拟引擎的 SiL Wrapper 模块和基于物理环境的 HiL Wrapper 模块, 保障所支撑的 CPS 软件运行能同时具备软件在环阶段的高可观测性、高可操控性, 以及硬件在环阶段的真实性。我们通过对大疆机甲大师软件运行过程的深

入分析, 提出了基于四向传感距离的定位技术和基于 API 中断的姿态调整技术, 支撑了 SiL Wrapper 模块和 HiL Wrapper 模块的高效准确交互, 确保了 CPS 软件执行的安全性、可观测性和可操作性。

本文组织结构如下, 首先介绍该测试床的服务对象大疆机甲大师机器人及其开发工具包; 然后展示了所提出的综合测试床的组织架构, 并着重介绍基于四向传感距离的定位技术和基于 API 监控的姿态调整技术这两项关键技术; 接着使用一个运行实例进行界面演示并总结全文。

1 大疆机甲大师软件开发工具

大疆机甲大师 EP 车 (RoboMaster-EP) 是大疆公司开发的一款教育型机器人, 具有强大的扩展性和可编程性。官方开源的软件开发工具包^[4](RoboMaster SDK)通过提供功能齐全的各类 API 降低了开发者的编程门槛, 因此在机器人竞赛和开源社区中广受欢迎。

图 1 左上角展示了用户在大疆机甲大师开发过程中, 需要设计并实现的机甲大师决策程序的一般模式: “程序周期性地检查小车状态(Status = Call 感知类 API), 并在小车满足对应条件时, 执行指定动作(Call 动作执行类 API)”。

RoboMaster SDK 实现了决策程序所需调用的两类 API。其中感知类 API 帮助程序获取环境信息, 如获取小车前向传感器读数; 动作执行类 API 发送指令对硬件进行操作, 如操控小车向前移动。

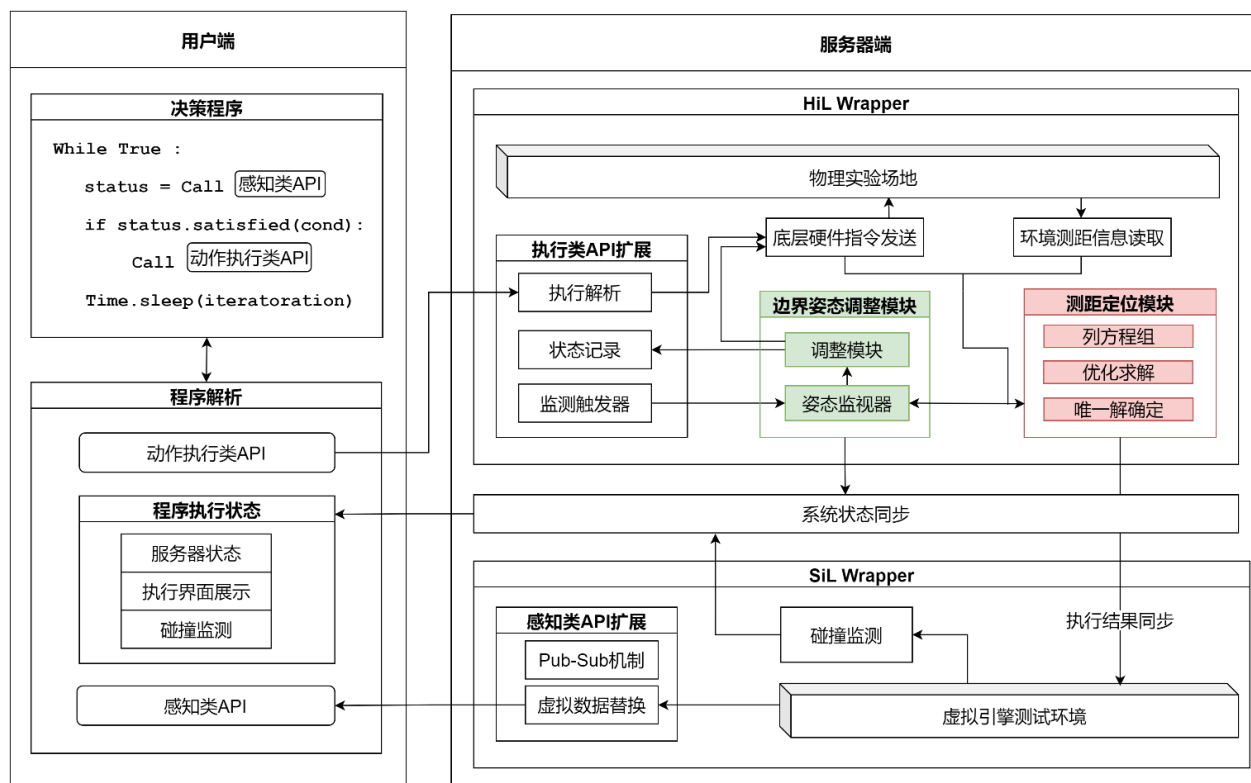


图 1 基于大疆机甲大师的信息物理综合测试床框架

在两类 API 的组合下，用户可以设计出功能繁多的控制程序，并在虚拟或现实的运行环境中进行开发。通常情况下，机甲大师程序首先会借助第三方虚拟引擎进行软件在环的开发，并在开发过程中逐步迁移至物理环境，最终实现硬件在环的开发过程。开发环境之间的迁移往往伴随着长时间的调试，高昂的搭建代价和运行时发生物理损坏的风险。

我们希望帮助用户平滑这一迁移过程，提供一种结合虚拟引擎和物理环境，介于 SiL 和 HiL 之间的综合开发平台。该开发平台需要具有虚拟环境搭建代价小、运行安全且易于调试的优势，同时保留物理环境中真实的软硬件交互过程。我们搭建基于大疆机甲大师的信息物理综合测试床实现了该目标。

2 基于大疆机甲大师的信息物理综合测试床

基于大疆机甲大师的信息物理综合测试床主体框架分为客户端和服务端两部分。

客户端负责对开发者执行的 CPS 软件决策程序进行解析，将程序中调用的 SDK 接口，根据其分类转发至服务器不同模块，并处理返回的执行结果。对于 CPS 软件程序而言，其与物理平台的交互过程将如软件在环一般获取虚拟环境中的环境输入，同时如硬件在环一般实现物理环境中的真实执行输出。

服务器端接受客户端用户程序中调用的 RoboMaster SDK API，并返回相应的执行结果。服务器端包含 HiL Wrapper 和 SiL Wrapper 两个模块，两

个模块分别负责在物理环境中执行被调用的动作执行类 API，和在虚拟环境中执行被调用的感知类 API。

更具体地，SiL Wrapper 模块通过第三方虚拟引擎搭建了虚拟环境，负责生成感知类 API 的执行结果并返回给调用方的决策程序。同时 SiL Wrapper 会接收定位模块产生的机甲大师定位数据，用于更新虚拟环境（如在虚拟环境中模拟机甲大师的移动过程），以便于产生 CPS 软件程序后续的软件在环环境输入。

HiL Wrapper 模块接受动作执行类 API 的调用，并与物理环境中的机甲大师交互，实现 CPS 软件程序的硬件在环执行输出。如何确保这一过程的安全性，并获取硬件在环运行的结果是实现所提出的信息物理综合测试床的关键。我们通过测距定位模块获取 CPS 软件程序的硬件在环运行的结果。该模块使用测距定位技术，在有界的正方形场地上，通过测距传感器数值联立方程组并求解，计算出小车位置信息。同时我们通过边界姿态调整模块保证硬件在环运行的安全性（如确保机甲大师不会与物理环境中的障碍物或边界发生碰撞）。该模块使用边界姿态调整技术对 API 的执行过程进行了优化，姿态监视器负责介入底层防止运行过程对物理环境造成损伤，调整模块处理因虚拟环境和物理场地不一致导致的动作执行受限问题。

下面我们详细介绍测试床的这两项核心技术：基于四向传感距离的定位技术和基于 API 中断的姿态调整技术。

2.1 基于四向传感距离的定位技术

基于四向传感距离的定位技术沟通了 CPS 软件程序在 HiL Wrapper 模块中的执行结果和 SiL Wrapper 模块中的模拟输入。更具体的, 定位技术需要实时测算并输出 CPS 软件程序控制下的物理平台(即大疆机甲大师)的位置信息, 并将其同步到虚拟环境中的虚拟引擎, 以产生后续的软件在环环境输入。

为此, 我们提出并实现了基于四向传感距离的定位技术。相比于一般定位技术, 如三角定位或图像识别, 该技术仅需使用机甲大师自带的四个红外传感器读数作为输入, 同时可以在面临环境非确定性时保持较高的定位准确率。

定位技术实时计算小车在物理场地中的位置, 输出的位置信息由小车坐标 (x_0, y_0) 和朝向 θ 构成。

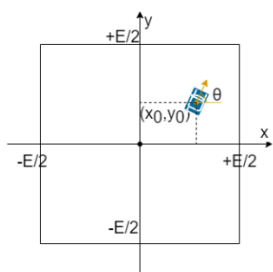


图2 小车位置信息示意图

三元组 (x_0, y_0) 和 θ 的定义如下: 已知物理场地是边长为 $E=2.4$ 米的正方形方型场地, 我们以方型场地的中心为原点, 作平行于边界的 x 轴和 y 轴。在该坐标系下, x_0 和 y_0 为小车中心点的坐标, θ 为小车前方朝向和 x 正轴的夹角。

定位技术的输入是小车前后左右四个测距传感器的测距值。在坐标系中, 正方形的四个边界可视为四条线段, 测距过程可视为过小车坐标 (x_0, y_0) 作平行或垂直于小车朝向 θ 的两条直线:

$$\begin{cases} y = -\cot \theta (x - x_0) + y_0 & (1) \\ y = \tan \theta (x - x_0) + y_0 & (2) \end{cases}$$

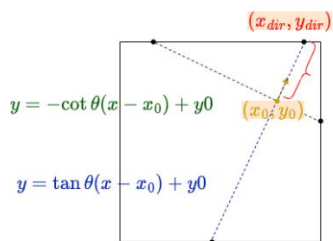


图3 测距直线和边界线段相交情况

由于小车在方型场地内部, 两条直线和四个边界线段会生成四个交点, 而前后左右四个方向上的测距值 $Dist_{dir}$ 等于该方向上交点坐标 (x_{dir}, y_{dir}) 到小车质心坐标 (x_0, y_0) 的欧式距离:

$$(x_{dir} - x_0)^2 + (y_{dir} - y_0)^2 = Dist_{dir}^2$$

如果已知每个交点所在的边界线段, 交点坐标 (x_{dir}, y_{dir}) 可以进一步通过 (x_0, y_0) 和 θ 表示, 这里, 我们使用枚举法列出交点情况, 并获得交点坐标的表示。

因此, 定位技术在获得四个测距值的输入后, 可以列出四个关于 (x_0, y_0) 和 θ 的方程, 再通过对该方程组的求解, 最终计算出当前测距值情况下的小车位置。上述过程会面临环境非确定性的影响, 为此, 我们通过将原方程组分成四组, 对每组的三个方程求得部分解后再进行合并, 实现了算法在输入含噪声情况下的求解。同时会使用历史位置信息和当前动作倾向对候选位置进行得分排序, 将得分最高的候选位置作为定位技术的输出。

2.2 基于 API 中断的姿态调整技术。

边界姿态调整技术包括边界检测技术和姿态调整技术, 前者是为了保证动作执行类 API 运行时的安全性, 后者是为了解决 SiL wrapper 开发场景和 HiL wrapper 物理场地不一致导致的 API 执行受限的问题。

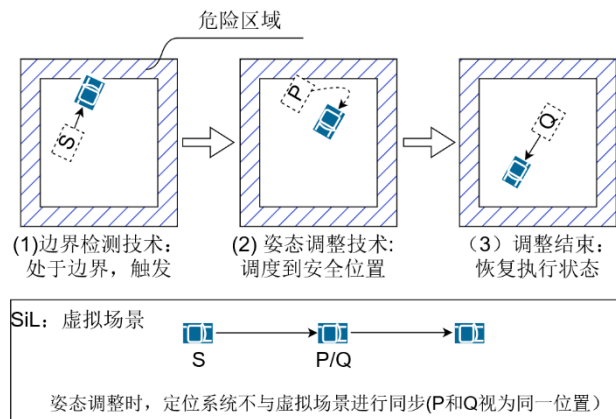


图4 平台视角下的检测和调整过程

缺少边界检测技术的原生动作执行类 API 运行期间可能会导致小车和物理环境中障碍物的碰撞, 造成硬件的损伤。边界检测技术通过设置触发器中断, 可以前瞻性地终止问题 API 的继续执行, 从而避免碰撞的发生。然而, 仅中断 API 运行会影响用户 CPS 软件程序的连续运行, 使得用户难以完整地测试其程序在所设置虚拟环境中的执行。

我们通过在 API 中断后加入平台控制的、对用户 CPS 软件程序透明的姿态调整过程来解决上述问题。图4 是平台视角下一次边界检测技术调整和生成执行结果的完成流程。首先, 边界检测模块将物理场地边界的临近区域(图4 阴影部分)设为危险区域, 小车到达该区域会触发中断条件, API 停止运行。

此时姿态调整模块接管小车, 首先调度小车到安全位置, 再通过中断前的信息恢复执行进度。在扩展

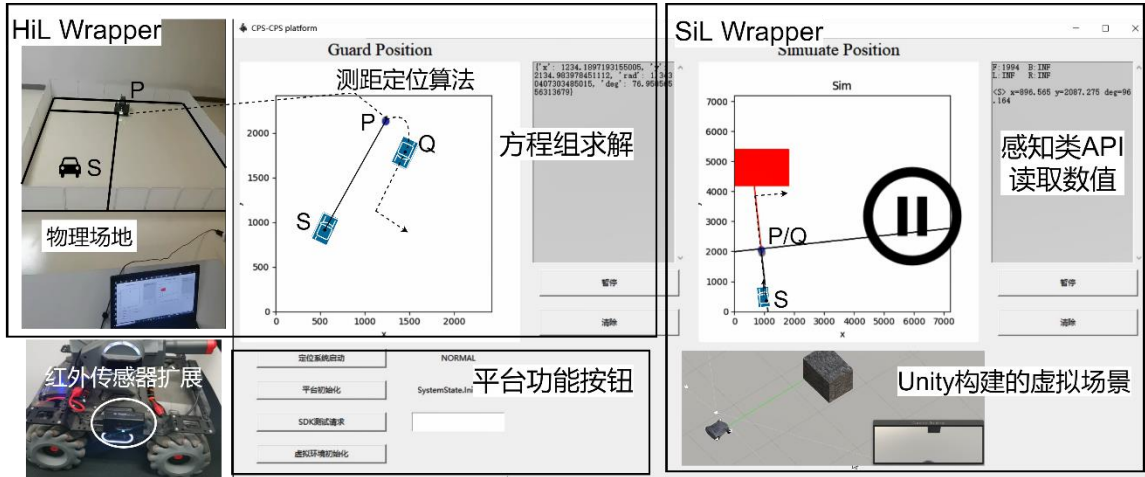


图 5 程序用例执行期间的界面截图

API 的运行期间,平台需要将调整阶段外的硬件执行结果向 SiL Wrapper 中的虚拟环境进行同步。

位(见图 5 左上),姿态调整模块在后台运行,负责控制小车执行。

为了实现上述姿态调整过程,我们在扩展的 RoboMaster SDK 中引入以下两个功能:

其一,设置平台运行时的 API 执行态和姿态调度态,定位系统将根据平台状态决定是否向虚拟环境映射执行结果,从而防止姿态调整期间的位置变动被记录。同时,平台需要对调整过程计时,为使用倒计时进行控制的执行类 API 补偿处于姿态调度态的时间。

其二,重构 SDK 接口代码,添加了监视进程、条件触发器、上下文记录等功能。该重构代码仅为服务器端使用,对用户而言修改应保持透明的,即用户在调用 API 时,不应该察觉到 API 行为的改变。这要求我们对阻塞和非阻塞两类动作执行 API 分别修改:(1)对阻塞类接口,SDK 在发送指令后睡眠,直到动作结束后才通过 Action 类唤起。我们使用 SafeAction 类提供 Action 类执行时等价的查询功能(执行进度、接口参数等),同时内置一个和该类生存周期一致的监测进程,负责中断触发和上下文切换。(2)对非阻塞类接口,SDK 仅负责发送指令,不做记录。我们需要添加全局观测者,对所有该类 API 的调用进行监视和中断触发,并记录最后一次的 API 调用以便恢复。

3 程序用例和界面展示

假设当前用户正在编写如下程序:“当小车前方测距值 $x > 0.2$ 米时,直行 $(x - 0.2)$ 米;当小车前方测距值 $x \leq 0.2$ 米时,设置前进方向为右”。并指定运行场景为空旷场地,距离小车前 3.5 米处有一障碍物。

该用例会涉及 4 次 API 调用时(见表 1)。对其中的感知类 API,我们在外部虚拟引擎 Unity 中搭建运行场景(见图 5 右下),用 SiL Wrapper 向外部引擎映射执行结果,并读取的场景中的测距信息。对于执行类 API,HiL Wrapper 通过测距定位算法求得小车定

表 1 用户示例在该场景下的 API 调用

调用 API 分类	API 描述	结果
感知类	①前方测距	3.48 米
执行类	②前进 3.28 米	位移 3.31 米
感知类	③前方测距	0.19 米
执行类	④设置前进方向为右	设置成功

当前是第一次调用执行类 API “②前进 3.28 米”期间的截图,此时小车前进动作过于靠近边界,触发边界检测中断,平台正处于姿态调整过程中。调整过程期间,由于 HiL Wrapper 不向 SiL Wrapper 进行同步,虚拟环境处于暂停状态。

4 总结

基于大疆机甲大师的信息物理综合测试床结合了 CPS 软件开发的软件在环阶段和硬件在环阶段各自的优势,为开发者提供安全、高可观测性、高可操作性的 RoboMaster 程序开发平台环境。

参考文献

[1] Lee E A. Fundamental Limits of Cyber-Physical Systems Modeling[J]. ACM Transactions on Cyber-Physical Systems, 2017, 1(1): 1–26.

[2] Zampetti F, Tamburri D A, Panichella S 等. Continuous Integration and Delivery Practices for Cyber-Physical Systems: An Interview-Based Study[J]. spanichella.github.io, 2021, 1(1).

[3] Butenuth M, Kallweit R, Prescher P. Vehicle-in-the-Loop Real-world Vehicle Tests Combined with Virtual Scenarios[J]. ATZ worldwide, Springer Science and Business Media LLC, 2017, 119(9): 52–55.

集成开源项目

[4] Dji RoboMaster-SDK: <https://github.com/dji-sdk/RoboMaster-SDK>

大疆机甲大师机器人开源官方 SDK 开发平台

项目发布地址

地址: https://gitee.com/KiLi_m/robo-master-sdk