用于可重复评估超宽带定位性能的自动化测试平台

Alexander Kemptner*†、Julian Karoliny*、Hannah Brunner‡、Andreas Gaich*、Michael Neubauer*、Fjolla Ademaj-Berisha*、Filippo Casamassima‡、Walther Pachler‡、Shrief Rizkalla*、Harald Witschnig‡、Andreas Springer†、Hans-Peter Bernhard*†

*Silicon Austria Labs GmbH，林茨，奥地利 (alexander.kemptner@silicon-austria.com)

†约翰内斯·开普勒大学，林茨，奥地利 ‡英飞凌科技奥地利股份公司，格拉茨，奥地利

arXiv:2603.03918v1 [eess.SP] 4 Mar 2026

摘要—**超宽带（UWB）系统的测试颇具挑战，因为多个设备需要通过有损链路进行协调，而且系统行为会受到时序、同步和环境因素的影响。传统测试往往不足以捕捉这些复杂的相互作用，因此亟需一种综合性测试平台基础设施，以管理设备、控制环境，并使测量和测试场景具备可重复性。本文介绍了一种高度自动化的测试平台架构，该架构基于机器人操作系统第2版构建，将设备管理与环境控制和测量系统集成在一起。该平台包括一个光学参考系统、一辆可控的自动导引车（用于在环境中定位设备），以及通过网络时间协议（NTP）实现的时间同步。该测试平台的定位可重复性均方根误差为 4.8 mm，朝向误差为 0.493°；基于 NTP 的同步方法实现了低于 1 ms 的时间准确度。所有测试平台功能均可通过简单的 Python 脚本远程控制，从而执行复杂测量场景等自动化编排任务。我们通过一项超宽带定位测量实验展示了这一点，说明该平台如何支持可重复、可观测且完全受控的无线实验。**

I. 引言

评估无线网络在超宽带（UWB）定位等应用中的性能和固件，需要具备多种功能。例如，(i) 需要为设备刷写更新后的固件，(ii) 需要控制电源和复位状态，以便正确初始化并开始测量，(iii) 必须收集日志，以便之后评估定位性能。此外，不仅需要管理设备，还应能够控制和测量环境。因此，测试平台的功能也应扩展至自动导引车（AGV）和光学参考系统等基础设施。对于超宽带系统而言，这些基础设施对于准确测量定位性能至关重要。与大多数测试平台不同，我们使用机器人操作系统第2版（ROS2）进行实现。这大大简化了测试平台的搭建，因为许多基础设施制造商都提供了现成的 ROS2 实现。此外，这还支持通过脚本或交互式用户界面集中控制整个测试平台。

文献中已提出了许多用于超宽带性能测量的测试平台。有一类测试平台并未将超宽带设备直接集成到测试平台架构中，而是使用嗅探器记录数据 [1]。这使得实验能够快速搭建测量环境，尤其适用于侧重数据后处理的实验。然而，如果需要测试许多不同的芯片或算法，手动协调设备（刷写固件、开始/停止测量等）就会耗费大量时间。

[2], [3]展示了更复杂、完全集成的超宽带测试平台配置。然而，这些配置也缺少设备固件更新等自动化功能，而这对于快速评估不同算法至关重要。 [4]中的测试平台使用机器人操作系统（ROS）（版本 1），通过集成式架构实现了自动化，但未提供超宽带评估所需的其他功能，例如位置真实值测量。 [5]中介绍的*Cloves* 测试平台也支持大规模刷写固件，但未集成自动导引车或参考系统。同样，[6] 中的测试平台采用定制架构，支持固件刷写和日志收集，但假设超宽带设备位置固定，也不提供实时位置真实值测量。 [7]所报告的 5G 测试平台和测量实验包含超宽带功能，其超宽带子系统采用了本文介绍的测试平台早期版本实现。 [8]使用了基于 ROS2、用于超宽带测量的完全集成配置。然而，作者侧重于测量结果，并未详细介绍测试平台本身的配置。总而言之，现有文献对完全集成的超宽带测试平台架构仅作了有限描述。

在本研究中，我们旨在通过完整描述这样一个能够对超宽带性能进行可重复、自动化测量的测试平台，填补这一空白。我们不只关注被测设备（DuT），还详细描述所有组件及其互连方式。我们基于开源实现开展工作，这大幅加快了有意复现我们设置的读者的实现进程。此外，我们还评估了测试平台本身的关键性能，包括其定位和朝向的重复性，以及其在有线和无线编排场景下的时间同步能力。最后，我们通过一个简单的基于网格的超宽带定位示例展示该系统，以彰显其端到端功能。

本文其余部分安排如下：第 II 节介绍测试平台的总体架构。第 III 节详细介绍该架构的硬件和软件实现，这是本文的主要贡献

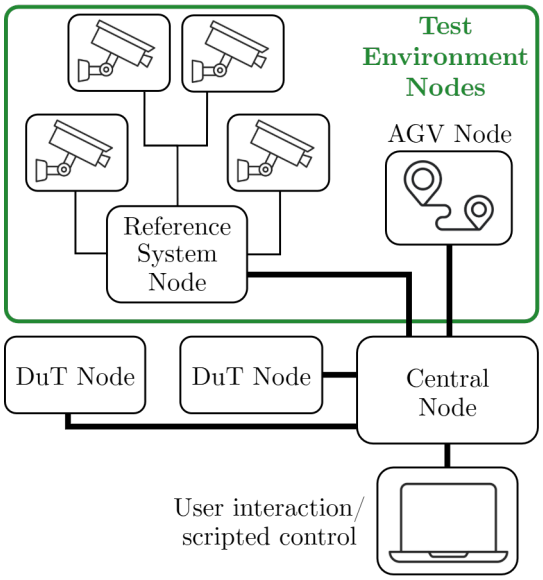


图1：测试平台的架构概览、其各个组成模块及其连接。

所在。第 IV 节对测试平台进行性能评估，并展示使用我们的装置进行超宽带定位测量的结果。

II. 测试平台架构

在本节中，我们重点介绍所提出测试平台的架构、各个组件以及它们的连接方式。在本研究中，该架构的组成模块被称为节点 。图1给出了概览，其中描绘了不同的节点类型。这些类型可分为被测设备节点、中央协调节点和测试环境节点。测试环境节点用于控制环境或与环境交互，例如参考系统节点或 AGV 节点。这些节点共同构成了我们全自动无线测试平台的架构基础。

A. 被测设备节点

测试平台的主要组件是搭载待评估相应固件的无线设备，我们将其称为被测设备（DuT）。在我们提出的架构中，它始终与作为管理设备的单板计算机（SBC）配对，由后者提供所有测试平台编排功能。二者共同构成一个被测设备节点。这样设计有两个原因。首先，被测设备的固件不会包含空中下载（OTA）更新等测试平台编排功能。其次，被测设备使用的无线信道不会受到干扰，因为管理设备可以使用与被测设备不同的通信技术。在我们提出的 UWB 测试方案中，可以通过 Wi-Fi 或以太网执行管理操作，而不会干扰 UWB 通信。固件和通信信道的分离确保测量仅评估被测设备的性能，而不受编排任务的影响。其主要功能包括

开启和关闭被测设备节点中的被测设备、与固件交互，以及收集测量结果或日志。

B. 测试环境节点

尽管被测设备是无线通信测试的主要关注点，但与环境的交互同样重要。在无线测试中，精确且可重复的测量只有在可控且可观测的环境中才能实现。因此，除被测设备节点外，我们的测试平台架构还纳入了测试环境节点。这一类别包括所有能够操控或监测环境的设备。操控的例子包括改变环境物体的位置，或调节光照和温度等条件。相反，环境状态测量包括物体跟踪或监测物理参数等活动。图1中的绿色框突出显示了我们测试平台中使用的两类此类节点示例。其中之一是 AGV 节点。其用途是为测量目的改变环境中物体的位置。例如，自动导引车可以移动被测设备节点，以评估其在不同空间配置下的行为。该架构对自动导引车的具体类型具有灵活性。它可以是地面机器人、无人机或其他车辆。第二个节点是用于记录位置真实值的监测节点。在我们的测试平台中，该参考系统节点管理一组高精度参考摄像机。

C. 中央节点与用户交互

测试平台的核心是中央节点，它连接所有各个节点，如图1所示。测试平台采用星型拓扑，以中央节点作为主要通信枢纽。它负责维持与各设备的连接，并协调各项任务，例如向其他节点发送控制命令以及记录测量。中央节点的另一项关键功能是向所有各个节点提供时间同步服务，从而确保整个系统拥有共同且一致的时间基准。除管理内部网络通信外，中央节点还充当主要用户界面。用户既可以通过交互方式向中央节点发送命令，也可以通过脚本化工作流与系统进行交互。后者可实现自动化且可重复的编排任务，例如开展复杂的测量场景。重要的是，用户仅与中央节点进行交互。发往其他节点（例如被测设备节点）的命令始终经由中央节点路由，由中央节点负责这些任务的协调，并将其分派给相应设备。

III. 测试平台实现

本节将详细介绍上述测试平台架构的实现。具体而言，我们将重点介绍图1所示各个节点之间的通信，以及用于实现这些通信的协议。为确保实现效率和可复现性，我们采用了标准且可公开获取的协议和中间件框架。具体来说，选择 ROS2

用于节点间通信，选择网络时间协议（NTP）实现节点间的时间同步，并采用表述性状态转移（REST）应用程序编程接口（API）与中央节点进行用户交互。选择这些技术，不仅因为它们在技术上适用，还因为它们是开放标准且得到广泛支持。这一选择为未来扩展奠定了坚实基础，也便于其他研究人员采用和复现该测试平台。

A. 使用 *ROS2* 进行编排

测试平台中的大部分通信都发生在节点之间。为此，我们使用 ROS2 [9]。ROS2 支持多种通信模式，提供了相应的软件工具，并且已有许多制造商提供集成方案，因此我们选择了它。ROS2 最初是一个用于构建机器人系统的软件平台。它提供了这些系统通常所需的大部分功能，还可以通过使用 C++ 或 Python 编写的自定义代码进行扩展，以适应特定系统的需求。虽然机器人系统与无线测试平台有所不同，但它们面临的许多挑战相似。两者都由相互连接的节点组成，这些节点通常运行在不同主机上，需要协同工作以实现共同目标。节点之间的互连需要借助节点发现和通信机制，而且通信可能跨越异构的有线和无线链路以及不同的操作系统。ROS2 提供了所有这些能力。其广泛应用也体现在许多设备制造商已提供代码，可将其产品集成到 ROS2 系统中，其中包括自动导引车和光学参考系统制造商。其他协议，例如 [7],所用的 MQTT，也能实现类似的编排效果。不过，ROS2 开箱即用提供的许多功能和协同优势，都需要手动实现。

1）网络与数据交换：在网络方面，ROS2 充当中间件，其提供的一项关键功能是点对点发现。这意味着节点无需知道其他节点的 IP地址，也能与它们通信。相反，它们可以通过名称寻址，也可以按类型枚举（例如，所有被测设备节点）。即使没有中央服务器，这一功能也能实现。因此，图1所示的中央节点并非 ROS2 的必需组件，而是我们在架构中有意采用中央编排中心的选择。消息可以通过三种不同的通信模式进行交换。如图2中被测设备节点旁以蓝色标示的内容所示，这些模式的特点如下：

话题遵循发布-订阅模式。这意味着一个或多个节点可以向特定主题发布消息。节点可以订阅该主题并接收消息。

服务 采用客户端-服务器模式。充当客户端的节点可以向特定节点发送请求，后者会以响应作答。请求方会等待响应，因此该节点需要及时作出答复。

动作 是话题和服务的组合。客户端节点向服务端节点发送请求。服务端节点启动一个长时间运行的任务，并通过初始响应通知任务开始。进度可以通过主题传回客户端节点。任务完成后，会发送响应。

无论是发现还是通信，ROS2都基于数据分发服务（DDS），这是一种开放的通信标准。它提供可配置的服务质量（QoS）参数，可用于根据可用带宽、允许的延迟或其他要求调整通信。例如，当QoS设置可靠性 设为尽力而为时，通过主题发送的传感器数值在发生通信错误时不会重传。同样，通过将QoS设置截止时间 设为该时间，若在指定时间内未收到任何数值，应用程序将收到通知 [9], [10]。除了灵活性之外，研究表明DDS协议在往返时间方面具有良好的性能 [11]。

2) 示例链路——中央节点到被测设备节点： 被测设备节点与中央节点之间的通信在图2中详细展示。如第 II-A节所述，被测设备节点由一个管理设备和一个或多个被测设备组成。它们通过USB串行通信连接，如图中绿色箭头所示。管理设备负责与测试平台的所有进一步通信。蓝色箭头所示的通信通过ROS2完成。ROS2提供的全部三种通信方案均被使用。串行日志以主题形式流式传输到中央节点，该主题支持多个节点对其进行订阅和发布。服务用于控制被测设备板的命令。最后，由于烧录完整固件是一项耗时较长的任务，因此将其实现为一个动作。

B. 时间同步策略

在自动化测试中，可靠的测量至关重要，而测量结果的有效性取决于对测量数据及其他相关事件进行精确的时间戳标记。此外，准确的同步还能加快测量活动，因为可以缩短各次测量之间的保护间隔。为实现尽可能高的准确度，时间戳标记直接在被测设备节点上执行，也就是在 UWB被测设备将信息发送至单板计算机时进行。随后，相应的时间信息通过 ROS2 日志主题转发至中央节点。与在中央节点进行时间戳标记相比，这种方法避免了 ROS2 通信引入的额外延迟和抖动，因此能够获得更精确的时间数据。由于 ROS2 本身不提供时间同步功能 [12], , 因此运行 ROS2 节点的主机系统使用 NTP 进行同步。相比精度更高的精确时间协议（PTP），我们更倾向于采用 NTP，因为后者要求所有节点通过以太网连接，并且需要特殊硬件支持 [13], , 这对于移动自动导引车和被测设备节点而言并不可行。完成同步后，ROS2 使用主机系统的时钟进行时间戳标记。在我们的架构中，NTP服务器运行于中央节点，所有其他节点则作为客户端与其同步。图2中的黄色箭头表示这一过程。正如第 IV-B 节的评估结果所示，即使通过 Wi-Fi 连接，该配置也能为测试平台提供足够的准确度。

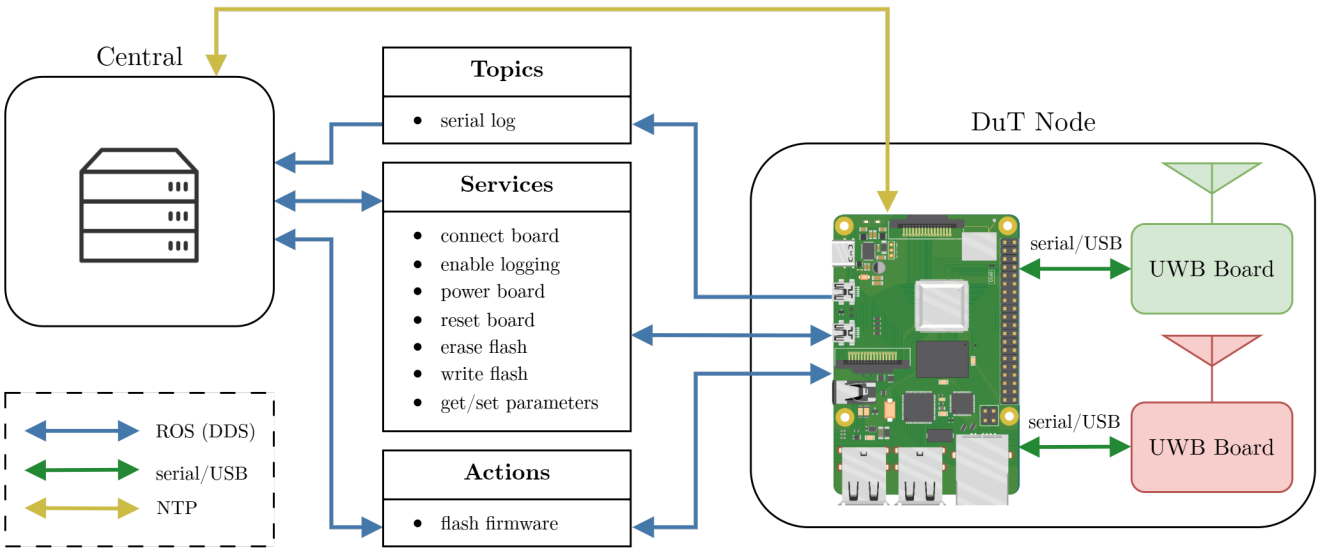


图2：详细框图　　中央节点与超宽带节点之间的信息和命令流。

C. 用户交互与自动化 *API*

中央节点与其他节点之间的所有内部通信均通过 ROS2 实现。用户仅通过中央节点与测试平台交互。中央节点负责处理命令，并按需转发命令。与节点之间的内部通信不同，用户与中央节点之间的交互基于 REST API。这种方式既能通过 REST 简便地与测试平台交互，又能通过 ROS2 实现对时间要求严格的协调。因此，用户设备无需安装 ROS2。此外，REST API 提供了定义明确的抽象层，可用于交互式控制和脚本化控制测试平台。具体而言，我们提供专用的交互式 Web 界面以及 Python API 作为用户界面。

如图2所示，该测试平台提供了丰富的功能来控制被测设备节点。用户可以控制被测设备的电源，通过打开或关闭 USB 端口实现。如果设备已通电，用户可以选择连接串行端口并启用日志记录。此操作会开始通过日志主题发送日志。用户可以使用 ROS2 提供的参数功能设置和读取串行端口、波特率等参数。用户可以通过重置固件、擦除闪存或刷写新固件来控制板上的固件。刷写新固件通过动作实现，因为这是一项耗时较长的任务。用户启动刷写后，中央节点会在 ROS2 请求消息中发送固件二进制文件，被测设备节点会通过响应确认收到。刷写完成后，被测设备节点会通过另一条响应通知中央节点。除了写入整个闪存外，也可以写入特定闪存地址，从而支持将唯一标识符等设备专属配置直接写入闪存。

测试环境也可获得类似程度的控制-节点。例如，自动导引车可以移动到特定位置和朝向，并提供内部里程计传感器数据，而参考系统节点则提供对真实位置测量值的访问和控制。

D. 软件与硬件配置

为了构建该测试平台，我们力求尽可能利用现有软件组件，以便轻松集成所提出的架构。这种做法凸显了 ROS2 的一项关键优势，即开源实现资源丰富。对于这些现有的基础组件和节点，我们不逐一列出通信端点，而是引用相应的开源项目。参考系统节点基于 MOCAP4ROS2 项目 [14]。AGV 节点使用制造商 Husarion 提供的 ROS2 软件包。为了在各节点之间实现基于 NTP 的时间同步，客户端（被测设备、参考系统和 AGV 节点）与服务器端（中央节点）均使用 *chrony* [15]。

专为本文所述测试平台设计和实现的主要组件是被测设备和中央节点。如图2所示，被测设备节点主要通过串行接口读取并转发日志，并将接收到的控制命令转换为 OpenOCD 命令。中央节点主要充当异步用户命令与其在 ROS2 域内分发之间的桥梁。如前所述，这一桥接功能通过 REST API 实现，该接口使用 FastAPI 框架构建。用于系统控制的交互式 Web 界面托管在中央节点上，并通过开源 Web 服务器 nginx 提供服务。

尽管所提出的架构在底层硬件方面具有灵活性，但我们在实际实现中选择了以下配置。每个

被测设备节点的管理部分由 Raspberry Pi 5 单板计算机构成，而被测设备本身则由 PSOC6 CYBLE-416045 评估板和 Quorvo DWM3000EVB 超宽带模块组成。参考系统由 OptiTrack 的十二台 Prime^{x} 22 摄像机组成，准确度为 ± 0.15 毫米。AGV 使用 Husarion 的 ROSbot XL。参考系统节点和中央节点均部署在运行 Ubuntu 24.04 的笔记本电脑上。所有组件的软件环境均使用 Docker 容器化，其中包括所有 ROS2 安装，以及 chrony 和 nginx 服务。这种方法确保了所有设备上的软件环境都可复现且保持一致。

IV. 评估：超宽带案例研究

本节评估所介绍的测试平台功能，并以室内网格超宽带定位作为应用示例。所有测量均通过测试平台的 Python 脚本接口完成。超宽带配置由实验室典型环境中天花板上固定且位置已知的 $N = 4$ 个锚点，以及一个放置在自动导引车上的标签组成。在测量脚本开始运行时，自动化测试平台会将这些不同角色对应的固件和配置直接烧录到相应的被测设备节点上。所有被测设备节点均工作在超宽带信道 9（7987.2MHz），带宽为 499.2 MHz。

在第一项评估中，我们研究自动导引车与光学参考系统相结合所能实现的定位重复性。随后，我们评估采用所述网络时间协议解决方案时，被测设备节点之间可达到的同步准确度。最后，我们通过执行一项简单的定位测量活动，展示测试平台的端到端功能。

A. 定位可重复性

为评估定位可重复性，我们在房间内选定一个固定的参考位置和朝向。自动导引车先驶向该位置和朝向，随后驶向 2x2m 网格内的随机位置和朝向，再返回参考位置。此交替模式重复了 100 次。对于每个位置，使用光学参考系统测量实际位置，并将其与目标位置进行比较。中央节点采用 PID 控制器，将自动导引车移动到设定位置，并使其误差保持在可配置的阈值范围内。尽管控制器也使用精度很高的参考系统作为位置输入，但由于机械因素，目标位置与实际位置之间通常仍存在差异

自动导引车的限制以及所选的最大允许速度。阈值设为 ± 0.5cm/$\pm 0.3°$ ，最大速度设为 0.1 m/s。由此得到的定位均方根误差（RMSE）约为 4.8 mm。另一项误差指标是 95%圆概率误差（CEP95），用于描述包含 95 % 测量值的半径。在我们的测量中，CEP95 约为 9.32mm。这表明，总体而言，该系统能够实现可重复的厘米级定位。同样，朝向的均方根误差约为 $0.493°$，这对于评估配备定向天线的被测设备尤为重要。

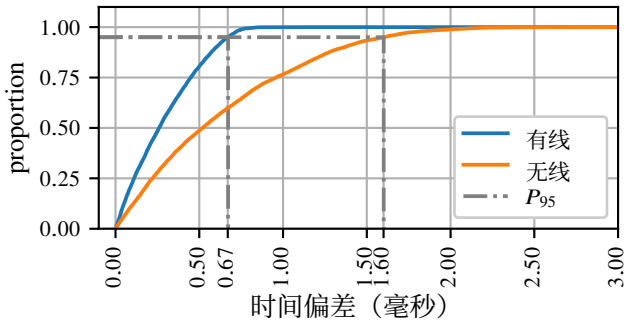


图 3：有线与无线链路上的网络时间协议同步准确度累积分布函数。

B. 同步准确度

为评估网络时间协议（NTP）同步的准确度，四个被测设备节点通过 5 GHz IEEE 802.11ac 无线链路连接到中央节点。此次测量未使用作为标签的被测设备节点。NTP 配置为每隔 16 至 256 s 动态同步一次。为了生成日志输出，我们将固件刷写到被测设备上，使其在通用输入/输出（GPIO）引脚检测到上升沿时通过串行接口写入数据。与测试平台的正常运行方式相同，此日志由被测设备节点上的单板计算机（SBC）收集并添加时间戳。测量期间，被测设备节点可同步 30 分钟。之后，任意波形发生器以 1 Hz 的频率同时触发 GPIO 引脚，持续 30 分钟。理想情况下，被测设备节点在每个上升沿生成的四个时间戳应当相同。然而，NTP 同步的不完善以及串行数据传输中的时序差异会导致偏差。我们通过比较每个上升沿时所有两节点组合的时间戳来评估这些偏差。结果以累积分布函数（CDF）形式显示于图 3。对于无线连接（橙线），95% 的节点偏差不超过 1.6 ms。时间偏差的均方根误差（RMSE）约为 0.81 ms。如第 III-B 节所述，大多数测试场景要求至少部分节点以无线方式连接，以实现移动性。然而，测试平台中的固定节点通常通过以太网连接。有线场景的性能由图 3 中的蓝线表示。95 % 的节点时间偏差不超过 0.67 ms，均方根误差（RMSE）为 0.36 ms。

C. 测试平台驱动的定位结果

为端到端展示测试平台的功能，我们开展了一次简单的测量实验。将充当标签的自动导引车上的被测设备节点移动到按 5x5 0.25 米网格图案排列的位置。在房间内不同位置、面积为 6×6米的区域中，共测量了五个此类网格。图 4 展示了其中一个网格的放大视图。光学参考系统测得并由参考系统节点捕获的真实位置以蓝点表示。每当到达目标位置时，便会发送一个

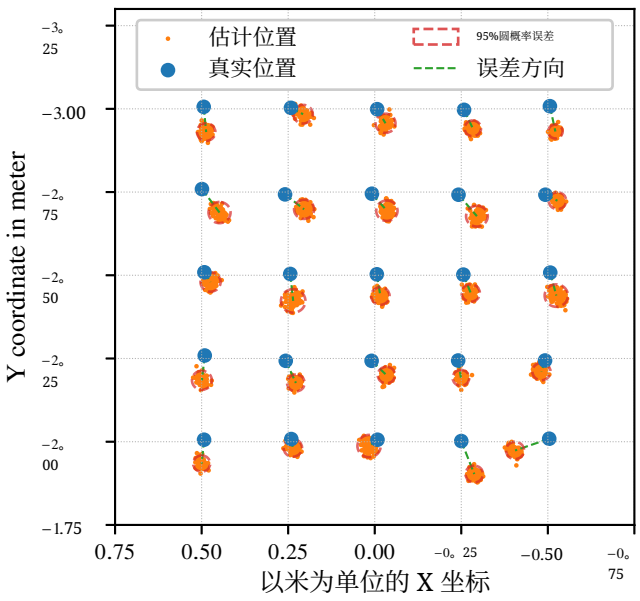


图 4：测量子网格视图。

重置命令，该命令会启动与每个 $N = 4$ 锚点进行 100 次双边双向测距 (DS-TWR) [16] 交换。使用测试平台的日志记录功能采集这些数据。基于这些距离测量值 d_i，我们通过求解非线性最小二乘问题来估计标签的位置 $\hat{\mathbf{p}} \in \mathbb{R}^3$

$$\hat{p} = \arg\min_{\mathbf{p}\in\mathbb{R}^3} \sum_{i=1}^{N} \left(\|\mathbf{a}_i - \mathbf{p}\|_2 - d_i\right)^2 , \qquad (1)$$

其中，$\mathbf{a}_i$ 表示第 i个锚点的位置。我们使用莱文伯格–马夸尔特算法求解。对于每个真实位置，由此得到 100 个估计位置，如橙色圆点所示。所有网格上的位置精度均方根误差为 14.56cm。红色圆圈表示每个位置的 95%圆概率误差，用于直观展示位置估计的精度。这些圆圈的平均半径为 5.09 cm。绿色线段连接真实位置与估计位置的中位数。线段长度直观表示位置估计的准确度。这些线段的平均长度为 9.39cm。

V. 结论与未来方向

本文介绍了一种用于超宽带性能测量的测试平台架构。该测试平台采用灵活的架构，涵盖所需的各个方面，例如对被测设备、参考系统和自动导引车的控制。灵活的架构使其能够以最少的重新配置部署到新场地。对于内部通信，采用了混合方案：使用 ROS2 处理对时效性要求较高的内部通信，并使用 REST API 提供用户界面。这两种通信协议都获得了广泛的开源社区和制造商支持，从而减少了实现工作量。由于 ROS2 不提供节点间的同步，因此部署并测试了网络时间协议（NTP）实现方案，结果显示，95% 的节点同步误差低于 1.6 毫秒，无线链路上也能达到这一水平。我们通过简单的定位测量对测试平台进行了评估，验证了其端到端功能。采用双边双向测距交换，与集成的光学参考系统提供的真实值相比，均方根误差为 14.56 厘米。最重要的是，该测试平台能够以很高的重复性进行自动化测量。只需再次运行 Python 测量脚本，即可快速比较更新固件或使用不同超宽带芯片时的性能。凭借这些特性，我们的测试平台将支持众多未来超宽带应用的开发、测试和最终普及，例如 FiRa Hero 用例 [17] 中的无跟踪导航、公共交通票款收取和资产跟踪。

参考文献

[1] P. Peterseil、B. Etzlinger、R. Khanzadeh 和 A. Springer，“超宽带室内定位的可信度评分”，载于 GLOBECOM，2023。[2] U. Raza、A. Khan、R. Kou、T. Farnham、T. Premalal、A. Stanoev 和 W. Thompson，“数据集：基于窄带、超宽带和动作捕捉系统的室内定位”，载于第2届从数据采集到分析研讨会论文集。ACM，2019，第34–36页。[3] M. Delamare、F. Duval 和 R. Boutteau，“基于超宽带和动作捕捉系统的工业场地人流新数据集”，《Sensors》，第20卷，第16期，第4511页，2020。[4] L. B. H¨ormann、A. P¨otsch、C. Kastl、P. Priller 和 A. Springer，“无线嵌入式设备分布式测试平台”，载于 IEEE 国际地中海通信与网络会议（MeditCom），2023，第381–386页。[5] D. Molteni、G. P. Picco、M. Trobinger 和 D. Vecchia，“Cloves：大规模超宽带测试平台”，载于第20届 ACM 嵌入式网络传感器系统会议论文集。ACM，2023，第808–809页。[6] M. Schuh、H. Brunner、M. Stocker、M. Schuß、C. A. Boano 和 K. R¨omer，“异构超宽带平台性能基准测试的初步探索”，载于 2022 年信息物理系统与物联网基准测试研讨会（CPS-IoTBench），2022。[7] D. Hamidovic、A. Hadziaganovic、J. Karoliny、A. Gaich、D. Klepatsch、M. Ahmed、R. Muzaffar、A. Springer 和 H.-P. Bernhard，“面向机器人协作的5G与超宽带融合”，载于 IEEE IECON，2025。[8] P. T. Mor´on、S. Salimpour、L. Fu、X. Yu、J. P. Queralta 和 T. Westerlund，“基于超宽带的无基础设施定位与多机器人相对定位基准测试：数据集与特性分析”，载于 2023 年 IEEE 传感器应用研讨会（SAS），2023，第1–6页。[9] S. Macenski、T. Foote、B. Gerkey、C. Lalancette 和 W. Woodall，“机器人操作系统 2：设计、架构及其实际应用”，《Science Robotics》，第7卷，第66期，第eabm6074页，2022。[在线]。可获取：https://www.science.org/doi/abs/ 10 1126 6074 . /scirobotics.abm [10] 2025 Open Robotics.（）服务质量设置。访问日期：2025年10月6日。[在线]。可获取：https://docs.ros.org/en/jazzy/Concepts/Intermediate/About-Quality-of-Service-Settings.html [11] S. Profanter、A. Tekat、K. Dorofeev、M. Rickert 和 A. Knoll，“OPC UA 与 ROS、DDS 和 MQTT 的比较：工业 4.0 协议性能评估”，载于 2019 年 IEEE 国际工业技术会议（ICIT）。[12] Tully Foote.（2021）时间同步方法。访问日期：2025年10月7日。[在线]。可获取：https://robotics.stackexchange.com/questions/99207/time-synchronization-approaches [13] P. Kulthumrongkul、S. Hengyotmark 和 W. Pora，“Raspberry Pi CM5 上精确时间协议与网络时间协议服务器的实现及性能分析”，载于 2025 ECTI-CON，2025，第1–5页。[14] ROSIN.（2025）MOCAP4ROS2-Project。访问日期：2025年9月30日。[在线]。可获取：https://github.com/MOCAP4ROS2-Project [15] M. Lichvar 和 R. Curnow。（2025）chrony。访问日期：2025年10月7日。[在线]。可获取：https://chrony-project.org [16] D. Neirynck、E. Luk 和 M. McLaughlin，“一种替代的双边双向测距方法”，载于 2016 年第13届定位、导航与通信研讨会（WPNC），2016，第1–4页。[17] FiRa 联盟。（ 2025 2025 03, ）FiRa年度报告。2026 [ ] 2026 02 访问日期：3月。在线。可获取：firaconsortium.org/sites/default/files/ - / FiRa 2025年年度报告.pdf

– – –