arXiv:2303.02340v1 [cs.RO] 4 Mar 2023

隐藏面向任务的编程复杂性：一项工业案例研究

Enrico Villagrossi[a]、Michele Delledonne[a, b]Marco Faroni[a]、Manuel Beschi[a, b]和 Nicola Pedrocchi[a]

[a]先进制造智能工业技术与系统研究所，意大利国家研究委员会，Via A. Corti 12，20133，意大利米兰；[b]机械与工业工程系，布雷西亚大学，Via Branze 39，25123，意大利布雷西亚

文章历史
编于2023年3月7日

摘要
在若干制造业领域，机器人编程界面的易用性构成了机器人应用推广的障碍，因为这需要最终用户具备更丰富的专业知识。目前的机器人编程方法大多沿袭过去，机器人程序员也不愿采用新的编程范式。本研究旨在评估向非专家用户引入一种新的 *task-oriented*编程界面所产生的影响，该界面隐藏了基于 ROS 的编程框架的复杂性。本文将这种界面的编程表现与基于先进机器人示教器的传统 *robot-oriented* 编程方法进行比较。一项实验活动邀请了 22 名非专家用户参与两项工业任务的编程。*Task-oriented* 和 *robot-oriented* 编程在学习时间、编程时间以及编程阶段提出的问题数量方面表现相当，凸显了向非专家用户顺利引入 *task-oriented* 编程的可能性。

关键词
直观机器人编程；面向任务的编程；人机交互；终端用户机器人编程；

1. 引言

1.1. 背景

过去几十年来，制造业中的工业机器人持续增长。根据《2021年世界机器人工业报告》（国际机器人联合会，2022年），工业机器人的两大主要应用是物料搬运和零部件装配，主要涉及抓取与放置任务。取放程序通常由一长串“移动到”指令构成。

本文已被 Taylor & Francis 出版的《International Journal of Computer Integrated Manufacturing》录用。联系EnricoVillagrossi。电子邮件：enrico.villagrossi@stiima.cnr.it

自机器人应用兴起之初，机器人程序的定义便基于示教式方法：程序员手动引导机器人沿着预期轨迹运动（引导式编程），或通过示教器进行操作（驱动式编程）。引导式编程常用于协作机器人。机器人结构轻巧且负载较低，因此可以直观地拖动末端执行器，使其在工作空间中移动；同时，机器人的设计可确保安全交互，并符合 ISO/TS 15066:2016（ISO/TC 299 机器人学，2016）等现行安全标准。

机器人编程技术的发展催生了面向机器人编程语言（Yang 等，2015）。它们是高级编程语言，主要基于 BASIC 和 PASCAL，例如 ABB Rapid、Fanuc Karel 和 Kuka KRL。这类编程语言集成了先进的机器人功能，可用于开发复杂的机器人应用，同时保留示教编程模式。不过，这些语言仍在不断发展，逐步融入编程领域其他部分的特性。其缺点是，每个机器人制造商都开发了自有的面向机器人编程语言，且彼此不兼容。掌握这些专有语言的基础知识所需时间较短。然而，深入掌握一个机器人平台需要丰富的经验和培训，因此机器人程序员往往专精于少数特定的机器人品牌。这种做法可能会使机器人程序员和机器人系统集成商不愿学习新的编程技能、了解不同的编程范式以及使用不同的机器人品牌。

目前，面向机器人的编程语言在工业应用中最为普遍，尽管离线编程环境正日益成为其补充。借助精准的机器人单元建模，在现场测试之前使用离线工具开发并仿真机器人程序，可以节省时间（Pan 等，2012）。将离线编程环境与 CAD 系统集成，也促成了机器人零件程序的自动生成（Castro 等，2019）。这类技术的典型用途是机加工、喷涂和电弧焊等连续过程（即，CAD/CAM 系统生成大量中间点，再由机器人进行插补）。对于给定的规划环境，离线编程可以部分生成无碰撞轨迹。更常见的用途是检查并标示机器人与环境之间是否存在碰撞。与面向机器人的编程语言一样，每家机器人制造商都开发了自己的离线编程环境，例如 ABB RobotStudio、Kuka.Sim、Motoman MotSim 和 Fanuc RoboGuide。不过，现场并非总能获得精准的三维模型，而且单元的建造也可能导致其与原始设计存在偏差。因此，在调试阶段，仍经常采用传统的手动引导编程方式，重新示教轨迹途经点。

尽管（i）机器人应用日益复杂，（ii）面向机器人 编程语言不断发展，（iii）离线编程工具得到采用，但多年来，机器人编程的方法以及机器人程序的组织结构几乎没有变化。机器人程序仍保持着僵化的结构：一系列指令经过编码后保存到机器人存储器中。此外，不熟悉机器人技术的终端用户也难以使用面向机器人的编程语言和离线编程工具。

ROS框架在机器人研究中的引入，推动了人们寻求一种面向标准化的编程方法，以促进最先进算法从研究领域向工业领域的技术转移。ROS采用可以

带来多项创新，例如面向任务的 编程、最先进的运动规划器、动态运动规划、仿真环境以及便捷的传感器集成。遗憾的是，尽管ROS-Industrial联盟付出了努力（ROS-Industrial联盟，2022年），ROS仍未能渗透到工业应用领域。目前，ROS-Industrial正转向ROS2，并再次尝试推动面向工业用途的ROS计划。ROS在工业领域推广的主要障碍，仍是机器人技术人员的学习速度较慢；这些人员不熟悉通用编程语言（GPL），例如$C/C++$、$Java$ 和$Python$，也不了解机器人框架的相关概念。要吸引不熟悉GPL的新用户，就需要改进编程接口，将ROS提供的先进功能与经典面向机器人的 编程语言的易用性相结合，并采用支持面向任务的 编程的设计。

1.2. 研究动机与贡献

尽管机器人研究领域已有多项成果展示，但先进机器人编程技术（如可视化编程或示教编程）在工业场景中的普及仍面临障碍（Villani 等，2018年）。主要障碍包括先进编程算法的鲁棒性、编程界面的复杂性，以及机器人程序员因熟悉文本式面向机器人 编程语言而不愿采用新编程技术所形成的技术惯性。

操作框架（MF）由 Villagrossi、Pedrocchi 和 Beschi，2021年提出，旨在推动当前工业机器人编程范式从面向机器人 范式转变为面向任务 编程工具，从而缩短编程时间，并简化复杂协作应用的开发。随后，MF 通过引入面向任务图形用户界面（下文简称 MFI）得到改进，该界面旨在隐藏基于 ROS 的框架的复杂性。本文旨在比较非专家机器人程序员和最终用户使用面向任务 界面编写工业任务程序时的接受程度、易用性和有效性。本文比较了使用 MFI 编程任务与使用机器人示教器（TP）进行经典引导式 编程的结果。测试所用设备为一台 UR10e 机器人及其示教器；如今，该示教器被认为是机器人示教器中直观性和易用性方面的先进代表[1]。

该研究涉及来自多个STEM院系的异质大学生群体，以及具有不同背景的机床操作员（作为最终用户测试人员）。只有少数测试人员此前具有少量机器人编程经验，并且没有人见过用于实验的MFI或示教器。

实验结果表明，使用作为MFI的面向任务的框架，学习时间可能会略有增加，但与标准编程技术相比能够带来多项优势。在需要处理以下情况时，这些优势尤为明显：任务重复、机器人重新编程和无碰撞运动规划。实验测试期间监测的、用于比较界面的参数包括学习时间、编程时间、编程过程中提出的问题数量、测试时间和重新编程时间。

两种编程方法的比较、在真实工业环境中开展工业任务的实验活动结果，以及

[1]标准工业机器人TP通常比UR的示教器复杂得多。

研究中有机床操作员参与（这在以往研究中并不常见），构成了值得关注的创新之处。

1.2.1. 论文大纲

本文结构如下：第2节分析相关研究，第3节介绍实验设置、实验设计、编程接口及所采用的方法。第4节讨论实验活动中获得的结果。最后，第5节总结结论并展望未来工作。

2. 相关研究

编程接口随着技术进步而不断发展（Siciliano等，2010年），为此提供了广泛的解决方案（Tsarouchi、Makris和Chryssolouris，2016；Mukherjee等，2022年；Heimann和Guhl，2020年）。然而，要让最终用户更容易进行机器人编程（Ajaykumar、Stiber和Huang，2021年），例如生产操作员，并推动机器人应用在新的制造领域中普及（Ajaykumar、Steele和Huang，2021年），仍存在差距。示教编程、可视化编程、增强现实/虚拟现实和自然语言编程等先进方法具有很高的直观性，因为它们属于自然和有形用户界面（自然用户界面和有形用户界面）。

示教编程是一种程序员可以向机器人示范任务的技术（Billard等，2008年；Zhang、Wang和Xiong，2016年）。示教编程的一种常见方法是发展引导式编程方法，利用导纳/阻抗控制算法，使用户通过机器人手动引导与机器人进行实体交互（Mohammad Safeea，2022年）。最常见的方法是在机器人法兰与末端执行器之间安装力/扭矩传感器（Bascetta等，2013年）。相反，现代协作机器人（如Kuka LBR或Franka Emika）则将扭矩传感器集成到机器人关节中。导纳/阻抗控制算法的缺点包括需要精确调节参数，而这可能引发稳定性问题（Ferraguti等，2019年）；需要额外的传感器（力/扭矩传感器）；并且对于大多数工业机器人而言，这类算法并未作为软件功能提供。此外，对于大多数工业应用而言，示范精度还不够高。

其他较不常见的方法则采用语音、视觉（Lu、Berger和Schilp，2022年）和动作捕捉（Makris等，2014年）。

可视化编程让非专家更容易上手编程（Coronado等，2020年；Huang等，2020年）。文献中常见的可视化编程界面包括流程图、行为树、程序块和图标（Stenmark、Haaae和Topp，2017年）。一些可视化编程系统提出了个性化的图形用户界面（Schou等，2018年），其中使用按钮、菜单、窗口、文本输入框和滑块。用于定义自动化编程语言的标准IEC-61131-3提供了可视化编程语言，例如功能块（FB）或顺序功能图（SFC）；这些语言也可作为标准语言用于机器人编程（Thormann和Winkler，2021年；Rendiniello等，2020年）。这些语言适合初学者使用。可视化应用的执行速度较慢，编程所需时间也比文本编程更长。复杂任务需要大量操作，用户得花时间为各项内容腾出空间、将其封装并重新编排到宏中，因此整个程序可能会变得拥挤。

增强现实/虚拟现实能够将现实环境与虚拟环境叠加。借助这项技术，一些信息或编程工具可以直接呈现在环境中。虚拟机器人与对象叠加后，操作员便可以使用示教编程，而无需直接与机器人交互（Blankemeyer等，2018年）。虚拟面板可以显示编程信息（例如，机器人轨迹或参数值），帮助操作员做出决策（Gadre等，2019年）。使用物理辅助工具（Ong等，2020年）或物体检测软件（Apostolopoulos等，2022年）可以方便用户进行编程。这项技术的实施成本较高；它需要具备更强的灵活性，因为它是针对特定任务进行编程的，而且在环境发生变化时容易出现故障。

自然语言编程使用语音和文本来创建程序。通常，这种技术会与另一种编程技术结合使用。人类语言的高度复杂性要求施加一定限制，而且这种语言无法完整描述动作。在相关文献中，自然语言编程与示教编程（Quintero等，2018年）、可视化编程（Huang和Cakmak，2017年）以及增强现实（Andronas等，2021年）相结合。在工业环境中，语音识别几乎不具备实用性；这些工作场所噪声很大，因此很难识别语音命令。

实体编程是一种利用实体对象定义程序结构的技术。通过在环境中放置特定对象，可以定义特定动作所需的对象、要执行的动作，以及执行动作的区域（Sefidgar、Agarwal和Cakmak，2017年）。另一种方法是使用特定卡片来定义任务结构；卡片的类型和顺序代表动作的类型和顺序（Kubota等，2020年）。对复杂任务进行编程可能需要大量对象，导致环境杂乱无章。有些动作难以通过对象描述，尤其是在需要较高精度时。

自动生成机器人零件程序的其他方法包括直接使用CAD模型提取控制机器人所需的信息，从而绕过机器人离线编程工具，如（Neto和Mendes，2013）所述。这种方法适用于连续流程。

欧盟项目 Robo-Partner（ROBO-PARTNER联盟，2013年11月—2017年4月）、Sharework（Sharework联盟，2018年11月—2022年10月）和 Sherlock（Sherlock联盟，2018年10月—2022年9月a）在各自范围内研究了先进编程技术在工业场景中的应用。Robo-Partner 面向装配任务的编程，采用了示教编程技术，包括音频命令、可视化编程，以及用户通过力觉/触觉传感直接操控机械臂，如（Michalos 等，2014）所述。Robo-Partner 的目标是让机器人技术更贴近中小企业，因为机器人知识匮乏是机器人应用的障碍。Sharework 旨在引入能够自主将机器人动作分配给机器人控制器的动态任务规划（Umbrico 等，2022）。该分配机制通过 ROS库 控制机器人，对协作任务的演进进行了评估（Faroni 等，2020）。该项目开发了基于示教学习的机器人运动规划算法，并提供了基于虚拟现实的直观人机交互方法。Sherlock 旨在为协作型和中高负载机器人提供零编程机器人技术。该项目开发了一种编程架构，将基于力传感器的增强现实（即，间接交互）与手动引导（即，直接交互）相结合。混合现实大致确定轨迹的起点和终点，而手动引导则可以对最终位置进行精细调整。

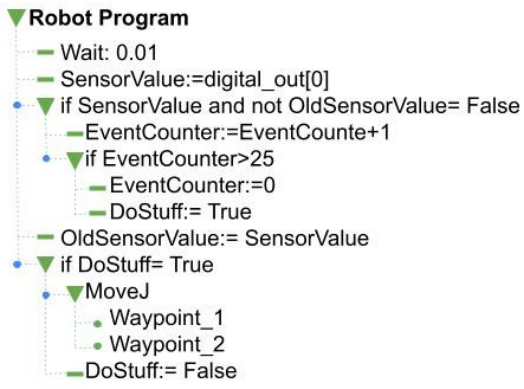


图1. 面向机器人的程序。

使用运动规划器和数字孪生可以生成无碰撞轨迹（Sherlock联盟，2018年10月—2022年9月b）。

目前有若干框架用于控制机器人，例如：CORBA（Schmidt、Levine 和 Mungee，1998）、YARP（Metta、Fitzpatrick 和 Natale，2006）、ROS（斯坦福人工智能实验室等，2022）、IMI2S（Anzalone 等，2014）。采用框架是机器人研究的特权；目前尚未观察到其在工业领域得到相关普及。其中，最成功的是 ROS，它代表了机器人研究编程的事实上的 标准。ROS 提供了先进功能和开源软件包，支持面向任务的 方法。另一方面，ROS 要求使用 GPL语言，而这类语言通常只在研究中使用[2]。

3. 材料与方法

3.1. 编程接口

本文比较了面向机器人的 编程界面和面向任务的 编程界面。面向机器人的 编程侧重于机器人能够执行的机器人基本运动。用户将这些基本动作组合成序列，以获得所需的程序。图1展示了一个面向机器人的程序示例。

面向任务的编程以任务为中心。用户通过设置工艺操作所需的参数来组合高级动作，而不是设置机器人的运动。用户无需从头定义基本动作，因为框架程序员已预先定义了任务结构。用户使用直观的语言编写代码，如图2所示。

实验中，每种类型都使用了一种编程界面。面向机器人的编程界面是 UR10e 示教器，而面向任务的编程界面则是一种基于操作框架构建的新型图形用户界面（Villagrossi、Pedrocchi 和 Beschi，2021年）。

[2]Kuka Sunrise.OS（Kuka AG，2022年）仅专用于 Kuka LBR iiwa，目前是一个独特的机器人操作系统示例，它提供基于 GPL语言（如 Java）的库，用于为机器人编程。

```
PLACE BEARING1 SO (SHAFT FITS BEARING1.HOLE) AND
                  (BEARING1.BOTTOM AGAINST SHAFT.LIP)
PLACE SPACER SO (SHAFT FITS SPACER.HOLE) AND
                (SPACER.BOTTOM AGAINST BEARING1.TOP)
PLACE BEARING SO (SHAFT FITS BEARING2.HOLE) AND
                 (BEARING2.BOTTOM AGAINST SPACER.TOP)
PLACE WASHER SO (SHAFT FITS WASHER.HOLE) AND
                (WASHER.BOTTOM AGAINST BEARING2.TOP)
SCREW-IN NUT ON SHAFT TO (TORQUE = t0)
```

图2. 面向任务的程序。



图3. UR10e 示教器。

3.1.1. 面向机器人的编程界面：*UR10e* 示教器界面

UR10e 示教器是一种高度直观的编程界面，基于不带实体按钮的触摸屏设备（除开关按钮和急停按钮外）；图3展示了 UR10e 示教器。示教器支持机器人在点动模式下移动、访问机器人配置和参数，以及通过面向机器人的高级编程语言对机器人进行编程。创建新的机器人程序时，需要依次添加移动指令，并通过示教机器人起始和结束配置来确定插补路径。机器人位置可通过引导式编程来示教，即使用所谓的手动引导模式移动机器人。程序员必须添加中间机器人配置（中间点），以确保生成无碰撞轨迹。示教器接口（TPI）提供了用于管理夹爪激活的专用功能。

3.1.2. 面向任务的编程界面：操作框架界面

MF 是一个专用于操作任务的软件包（Villagrossi andBeschi, 2022a,b）。MF 提供了一些基本组件，使面向任务的机器人编程成为可能。图 4 展示了 MF 的结构。对于给定的操作任务，MF 旨在：(i) 处理操作动作（例如抓取、放置、移动到），使程序员不必管理和执行各个动作；(ii) 处理机器人系统（机械臂 + 抓取系统）的运动学模型，并能够计算正运动学和逆运动学；(iii) 集成运动规划功能，以便针对给定的规划场景，在规划环境中的给定机器人系统上生成无碰撞轨迹，而规划环境可以动态变化；(iv) 执行

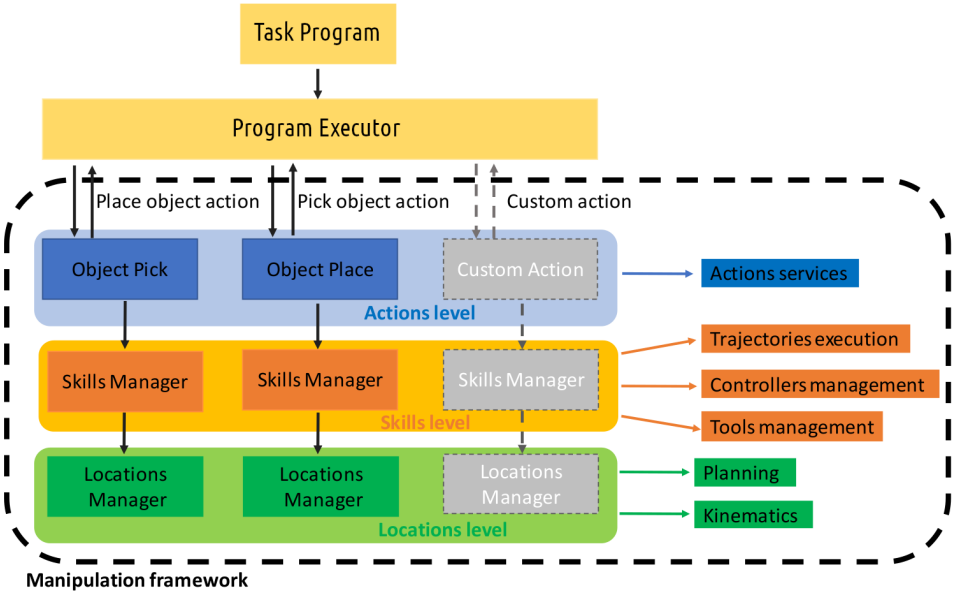


图4. 操作框架层级说明。

在所需机器人系统上规划的轨迹；(v) 动态更改机器人控制器。运动规划使用 *MoveIt!* 流水线（开源社区，2022年）。*MoveIt!* 支持将运动规划器动态加载为插件，从而针对给定的规划场景实现无碰撞运动规划。更多信息请参见（Villagrossi, Pedrocchi, andBeschi,2021）。

MF的API需要*C++* 编程和ROS集成（例如，创建ROS节点、定义配置文件以及数据集管理）。为克服这一限制，研究人员开发了一个图形用户界面，称为MF界面（MFI）（Delledonne和Beschi，2022a,b）。MFI采用面向任务的编程结构。该界面允许用户轻松串联多个动作，以组装一个任务。用户可以通过与机器人交互来设置所有组件，从而专注于任务定义。用户可以在预编程的动作（如抓取、放置、前往）之间进行选择。如图5所示，程序可以通过拖放所需的动作来组成。每个动作由一组参数定义。例如，一个抓取动作需要接近位姿、抓取位姿和夹爪闭合力（见图6b）。需要注意的是，MFI无需指定中间点，因为MF会根据任务规范自动规划轨迹。

MFI 的一个相关功能是允许指定一个对象类型，将多个对象归为一组。在运动规划过程中，MF 会自动计算同一对象类型 中每个对象的抓取位姿（以及逆运动学解）。随后会选择路径规划最优的对象。

将机器人移动到所需位置，即可获取机器人位姿，这与经典的 TP 编程类似。MFI 支持手动引导 编程（前提是机器人配备了力/力矩传感器）和遥操作，见图 7。MFI 可轻松连接视觉系统，自动定位机器人工作空间中的对象位置，从而提高编程的灵活性和自主性。

该界面支持执行动作或动作 序列。第一步是进行运动学计算（即 任务中涉及的机器人和对象），随后计算无碰撞运动规划。错误消息会通知操作员

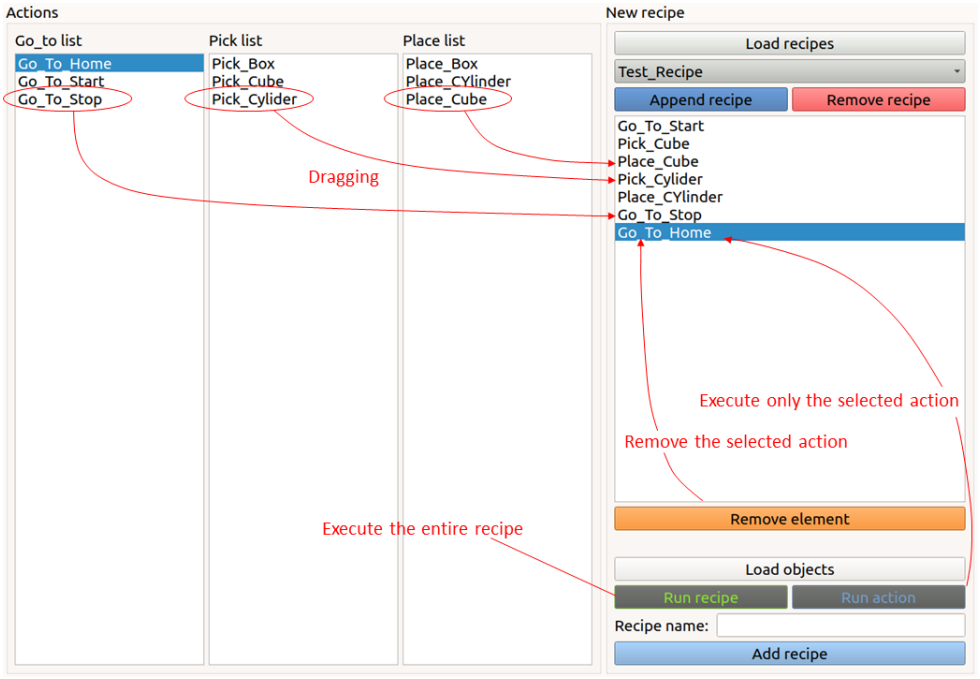
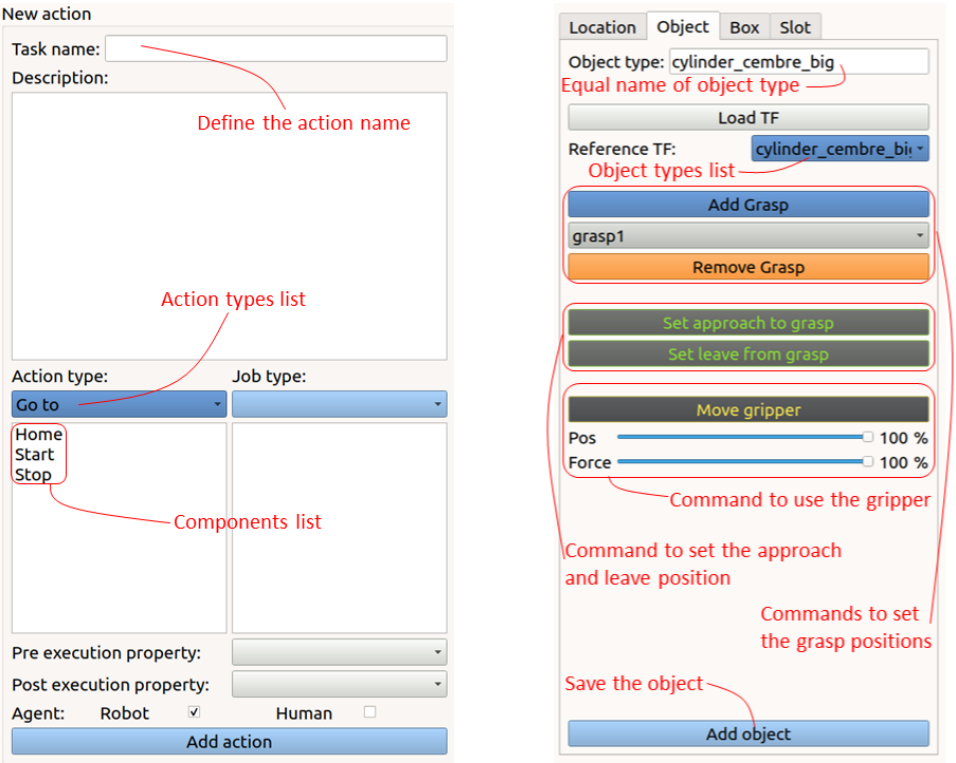


图 5. MFI：通过串联多个任务动作来组合任务的示例。



(a) 动作设置　　(b) 对象设置

图 6. 动作设置和组件设置面板。

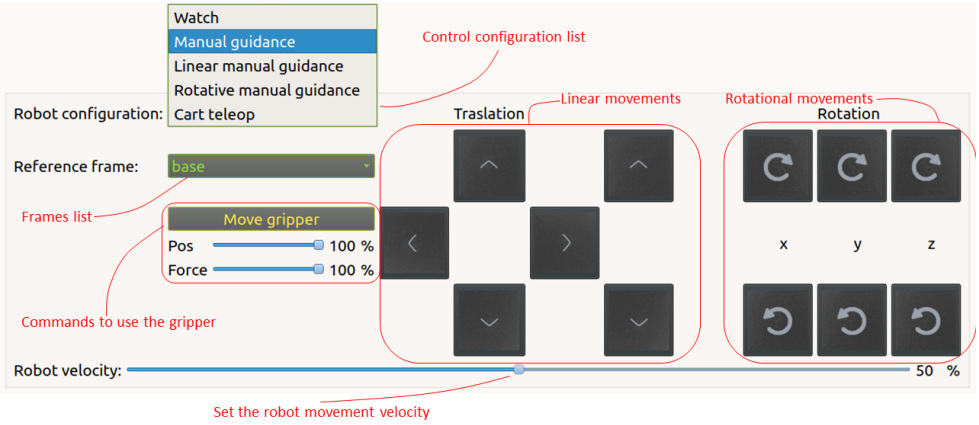


图 7. 接口操纵杆，通过相对于所选参考坐标系移动末端执行器坐标系来执行运动。

如果逆运动学或运动规划不可行，则可以修改组件或动作 序列来解决问题。目前，该框架仅提供三种动作。开发新动作仍需要专家开发人员；然而，解决复杂工业任务所需的动作数量是有限的（Bøgh 等人，2012），而且 MF 中提供的动作使其能够覆盖广泛的任务。从基本技能开始直观地生成自定义动作的功能将作为未来工作的一部分。

3.2. 实验设置

实验设置由安装在 Cobo-tracks 线性导轨上的 Universal Robots UR10e 组成。机器人夹爪为 Robotiq 2F-85。一台外部个人计算机以 $500[Hz]$ 的频率控制机器人。个人计算机与机器人通信所使用的软件是官方 ROS 软件包（Universal Robots，2022）。该软件包需要经过微插值的关节位置来控制机器人。反馈信息包括关节位置、速度、电流和外力。

机器人的末端执行器为一款 Robotiq 夹爪，带有两个夹指，工作范围为 $85[mm]$；力/力矩传感器安装在机器人法兰与夹爪之间。力测量范围为 $100[N]$，力矩测量范围为 $10[Nm]$。图 8 展示了实验设置，并突出显示了吊顶式机器人配置。实验期间未使用 Cobotracks 线性导轨。

3.3. 实验描述

在实验过程中，用户分别利用以机器人为导向的（即UR10 示教器界面，以下简称 TPI）和以任务为导向的（即MFI）编程接口编写了两项任务。任务1是一项包含 10 个对象的抓取与放置任务，旨在通过高度重复性的任务衡量两种编程接口的性能（即编程和测试时间）。任务2模拟了一个机床上下料应用场景，其中机器人必须进入机床工作空间，在避开障碍物的同时取出或放置工件。该任务旨在衡量当需要处理机器人工作空间中的多个障碍物、受限空间以及重新编程时，这些界面的性能。

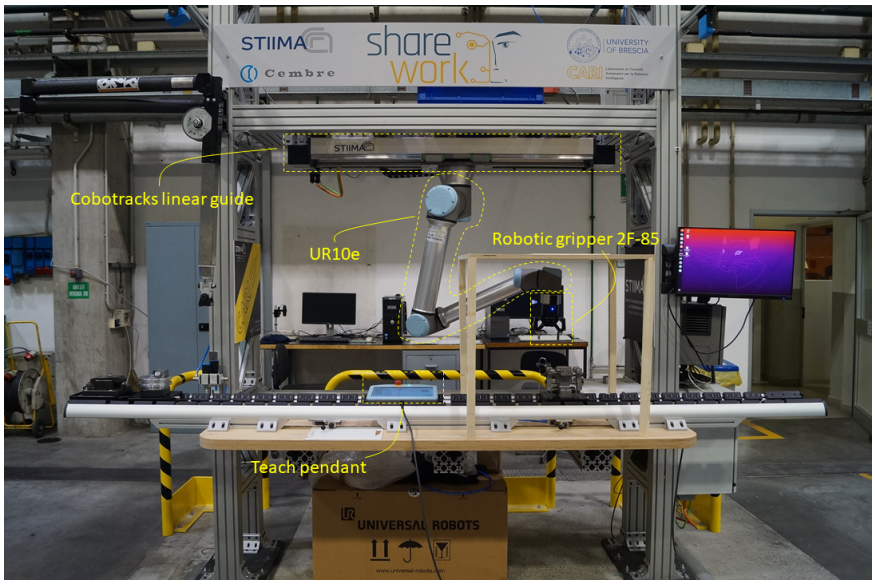


图 8. 实验设置。

实验包括五个阶段：

(1) 介绍：向用户介绍实验和测试阶段。(2) 教学：用户观看一段视频，了解界面（即 TPI机器人或操控框架）及其用法。随后，专家操作员会在辅助训练中为用户提供支持，训练目标是完成一次抓取与放置。在此阶段，用户可以随时向培训员提问。训练会持续进行，直到用户表示自己能够自主编程任务。最后，专家操作员向用户说明需要编程的任务；随后即可开始编程阶段。(3) 自主编程：用户在没有专家帮助的情况下为机器人编程。当用户表示已完成任务编程时，此阶段结束。如果用户无法继续进行任务编程，可以提问。问题数量是一个评估参数。(4) 测试：测试在上一阶段开发的程序。如果任务正确执行，此阶段即告结束。反之，用户必须对程序进行修正和测试，直至程序完成。任务是否成功决定了此阶段的结束。(5) 问卷：用户填写一份关于界面直观性和复杂程度的问卷。

任务1和任务2的实验由22人参与完成。所有受试者均自愿参与，并根据《赫尔辛基宣言》签署了知情同意书。

3.3.1. 任务*1*实验描述

任务1要求抓取与放置十个对象。这些对象分为两种类型（立方体和圆柱体），每种各有5个。抓取后，机器人根据对象类型将对象放入底座上相应的箱子中。程序员可以自行决定抓取顺序，并在必要时确定轨迹途经点。图9展示了对象和箱子的布局。右侧的箱子用于放置立方体，左侧的箱子用于

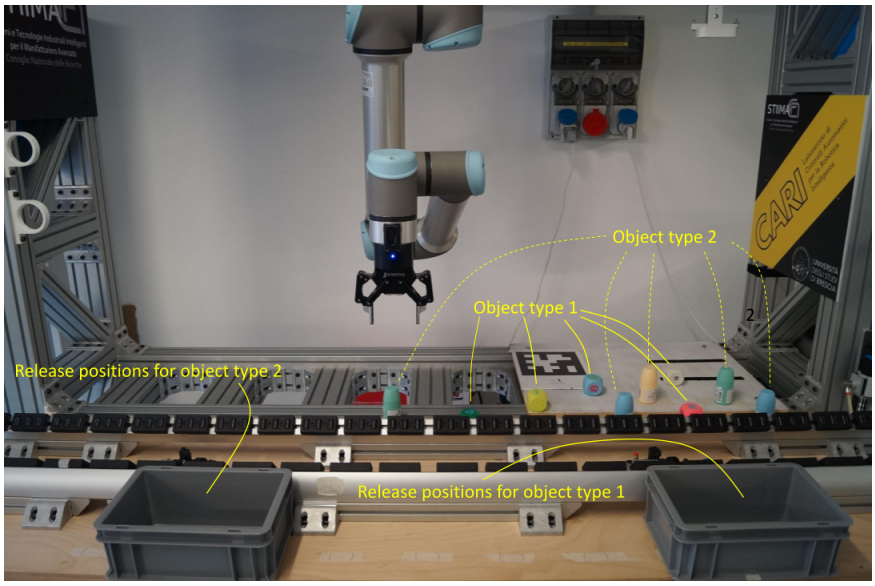


图9. 任务*1*的设置与说明。

放置圆柱体。

实验招募了两组用户，以下分别称为A组和B组。用户大多是来自STEM院系、年龄在20至35岁之间的大学生。A组由8人组成，他们使用第3.1.1节中介绍的UR10e 示教器对应用进行编程。B组由9人组成，他们使用第3.1.2节中介绍的操控框架对同一应用进行编程。

实验由第3.3节所述的五个阶段组成。

3.3.2. 任务*2*实验说明

任务2要求从受限箱体中抓取两个对象，将其移出箱体并放置在指定区域。抓取顺序不限；程序员可以自行决定抓取位置。程序完成并经过测试后，程序员必须更改对象的释放位置。

这样做是为了避免在需要大量重复动作 （如任务1）时对TPI造成不利影响。事实上，该任务仅涉及两个对象。同时，机器人必须穿过一个有障碍物的受限环境，以评估操控框架的规划性能（即 无碰撞运动规划），并与示教器进行比较；使用示教器时，程序员必须通过添加中间轨迹途经点来处理避碰问题。

该任务模拟了工业机器人在机床上下料中的典型应用场景：机器人需要进入机床工作空间，取出或放入工件，而多个障碍物限制了规划环境。程序员必须谨慎处理可能发生的碰撞。由于机器人单元发生变化，重新编程机器人任务是常见需求。实际上，在任务编程结束时，我们要求对机器人程序进行少量调整，以应对机器人单元的变化。我们要求使用两种界面，部分修改原有机器人程序，并测量重新编程时间。当生产具有小批量和极大的生产多样性时，这类需求十分常见。

T实验在一家机械工程公司的车间进行 m-

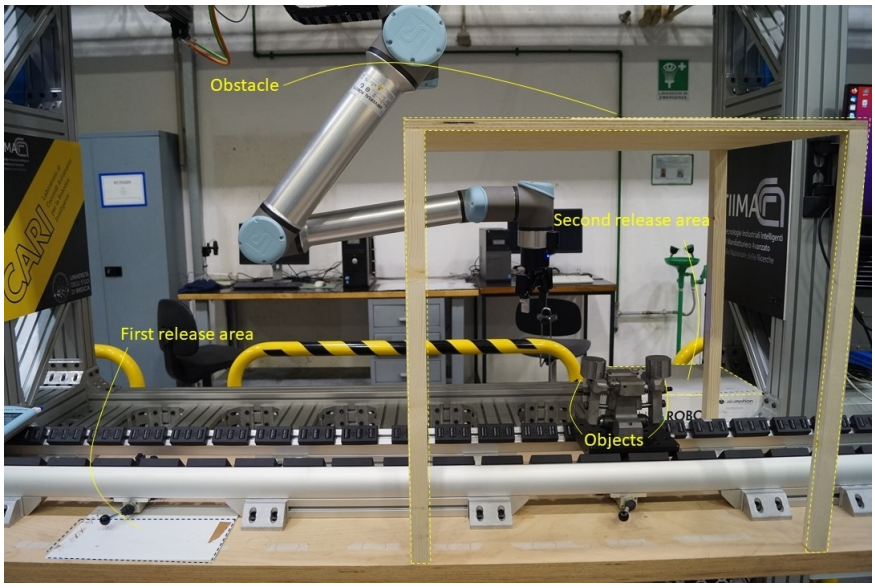


图10. 任务2的设置与描述。

。参与实验的是机床操作员和一名生产工程师，他们既没有机器人背景，也没有编写机器人程序的经验；这是他们第一次接触这两种编程界面。用户组由5人组成，年龄介于26至46岁；其中4人高中毕业，第5人拥有自动化工程学位。所有用户都有技术背景和机床编程经验，但没有工业机器人使用经验。

主办公司提供的用户数量有限，因此无法组建两个独立组；所以，所有参与者都使用两种界面各重复进行了两次实验。第一次实验可能会影响第二次实验的学习阶段，因为编程经验可能会让参与者对第二次实验的任务有所了解；界面的使用顺序随机交替，以避免任何可能的偏差。参与任务2实验的用户没有参与任务1的实验，反之亦然。

该实验由第3.3节所述的五个阶段组成，并在第4阶段之后增加了一个自主重新编程阶段，在此阶段用户需要修改已开发的程序。记录用户在自主重新编程过程中花费的时间以及问题数量，作为评估参数。增加了第二个测试阶段，以纠正任何重新编程错误。最后，用户需要填写问卷。

3.4. 性能测量

一般而言，在任务编程过程中，可以确定 4 个主要阶段：(i) 用户学习阶段，(ii) 程序开发阶段，(iii) 测试阶段，(iv) 执行阶段。第一个评估指标是这些阶段所用的时间。因此，选择了以下评估参数：

- 学习时间（$\boldsymbol{LeT}$）：用户在第 2 阶段（教学）进行辅助编程期间学习编程环境所花费的时间。LeT 用于评估

表1. 任务1和任务2结束后向用户提出的问卷。

1	请用1到10分评分，其中1分表示缺乏经验，10分表示经验非常丰富： 您使用工业机器人的经验有多丰富?
2	请用1到10分评分，其中1分表示不简单，10分表示非常简单：您觉得 使用建议您使用的编程界面有多简单?
3	请用1到10分评分，其中1分表示不直观，10分表示非常直观：您觉得 您觉得所提供的编程界面直观吗?
4	请按 1 到 10 分打分，1 表示不快，10 表示非常快。您觉得 学习使用所提供的编程界面有多快?
5	请按 1 到 10 分打分，1 表示毫无帮助，10 表示非常有帮助。您的先前 知识对您学习使用所提供的编程界面有多大影响?

界面学习曲线的陡峭程度。

• 编程时间（编程时间）：用户开发机器人程序以完成给定任务所花费的时间。编程时间 反映了界面的易用性和直观性。

• 测试时间（$\boldsymbol{TeT}$）：用户测试机器人程序所花费的时间。程序员必须在真实工作环境中测试开发的程序，检查是否可能发生碰撞，并在需要时添加或调整中间点。TeT 是衡量界面在任务编程过程中避免用户出错能力的一个替代指标，因为它反映了程序调试所花费的时间。

• 执行时间（$\boldsymbol{ExT}$）：机器人执行程序所花费的时间。执行时间与机器人速度（两个界面下的速度相同）以及运动规划器找到的路径长度有关。使用 TP 时，ExT 取决于程序员选定的中间点数量及其位置；使用 MFI 时，轨迹长度由选定的运动规划器决定。

• 执行测试次数（$\boldsymbol{TeN}$）：成功执行任务前运行的次数（不包括ExT）。与TeT类似，TeN 用于评估编程界面对用户在编程过程中所犯错误的鲁棒性。

•编程期间的问题数量（$\boldsymbol{PrQ}$）和测试期间的问题数量（$\boldsymbol{TeQ}$）：问题数量体现了用户对界面使用方式的理解程度，可用于评估界面的易用性和直观性；

在任务2的实验期间，还测量了以下参数，以评估重新编程阶段：

• 重新编程时间（$\boldsymbol{ReT}$）：与重新编程阶段的PrT相同。

• 重新编程测试时间（$\boldsymbol{ReTeT}$）：与重新编程阶段的TeT 相同。

• 重新编程执行时间（$\boldsymbol{ReExT}$）：与重新编程阶段的ExT 相同。

• 重新编程期间的测试次数（$\boldsymbol{ReTeN}$）：与重新编程阶段的TeN相同。

• 重新编程期间的问题数量（$\boldsymbol{ReQ}$）和重新编程测试的问题数量（$\boldsymbol{ReTeQ}$）；

最后，为了评估用户感受，用户填写了表1中的问卷。

4. 结果与讨论

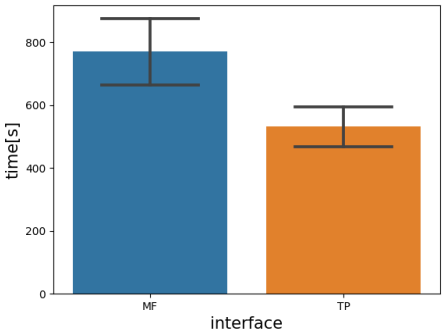
实验执行过程中采用的方法见第3.3节。评估使用了第3.4节中介绍的性能指标。

4.1. 任务*1*实验结果

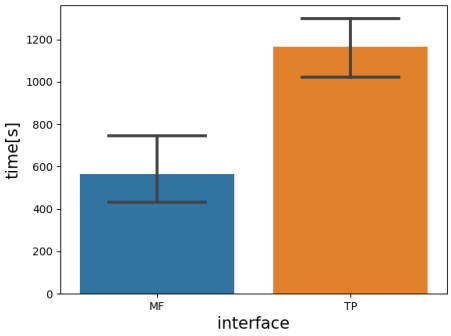
任务1的实验采用了第3.3节和第3.3.1节所述的方法。第3.4节报告了用于评估界面的测量参数。任务1的所有评估指标均列于图11和图12中。

MFI的平均学习时间 LeT 比TPI高44.9%，见图11a。正如预期，MFI的概念比TPI更复杂，因此学习时间往往更长。MFI的平均编程时间PrT比TPI低51.7%，见图11b。TPI的PrT 值较高，直接与任务涉及的对象数量有关，因为操作员需要示教多个位置，才能实现无碰撞轨迹。相比之下，使用MFI定义抓取与放置任务$Pick\&\ Place\ Task$ 时，只需示教两个抓取位置即可定义抓取动作 ***Pick Action*** ，再示教两个释放位置即可定义放置动作 ***Place Action***。定义为动作 ***Action***后，操作员便可重复执行相同的动作 ***Action***。两种界面的平均编程问题$PrQs$ 和测试问题$TeQs$数量都较少，分别见图11c和图11f。这些较低的数值表明操作员学习率较高，也说明两种界面的易用性相当。测试次数$TeNs$ 和测试时间TeT也较少，分别见图11e和图11d。大多数实验在编程过程中没有出现错误。错误较少，因此许多实验无需纠正；需要纠正时，纠正时间也很短，这进一步印证了PrT 值所体现的结果。错误数量较少，使$TeTs$ 与$ExTs$重合。TPI的平均执行时间ExT 低13.5%，见图11g。机器人运动规划器会对通过TPI示教的轨迹中间点进行插值。轨迹计算耗时较短，且中间点采用线性插值。相比之下，MFI会以最优无碰撞轨迹对起始位置和目标位置进行插值。评估规划场景并生成无碰撞轨迹所需的计算时间可能有所不同，这解释了执行时间的差异。尽管如此，结果之间的差异并不显著；此外，TPI的ExT标准差大于MFI。TPI的ExT 标准差较大，说明其对用户技能的依赖程度较高。MFI的ExT 标准差较小，因为执行过程不受操作员能力的影响。

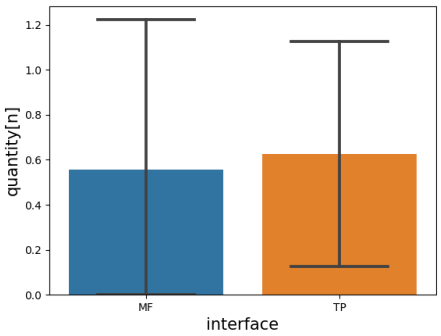
最后，研究向用户发放了一份问卷（见表1）。问题涉及用户对自身既往经验水平（Q1）、易用性（Q2）、直观性（Q3）、学习速度（Q4）以及先前知识对于编程任务的效用（Q5）的看法。图11h、图12a和图12b显示，在易用性、直观性和学习速度方面，MFI与TPI的结果几乎相同。由于问题1和问题5（用户的既往经验水平，以及先前知识对任务编程的效用）的得分始终较低，因此未报告这两题的得分图。由此可见，其信息贡献并不显著。表2a和表2b展示了参与任务1实验的用户信息。



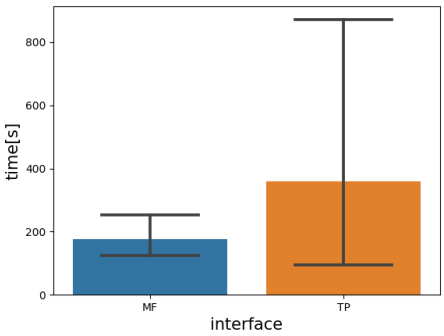
(a)学习时间（*LeT*）。



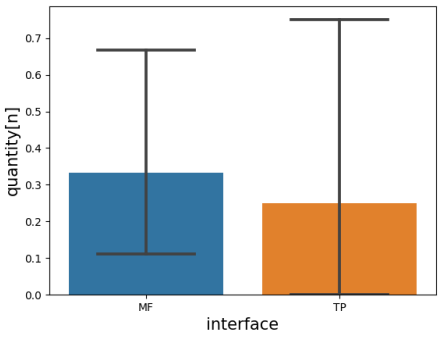
(b)编程时间（*PrT*）。



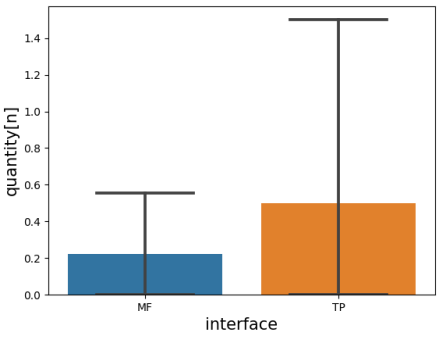
(c)编程问题（*PrQs*）。



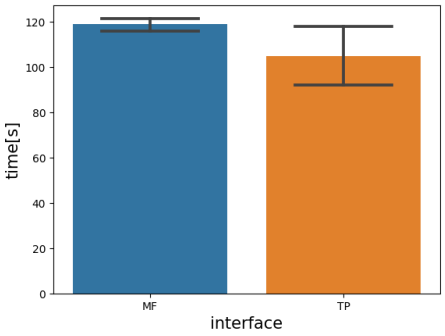
(d)测试时间（*TeT*）。



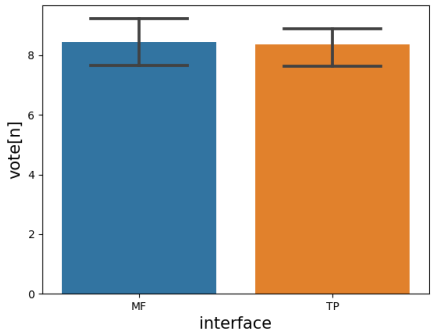
(e)测试数量（*TeNs*）。



(f)测试问题（*TeQs*）。

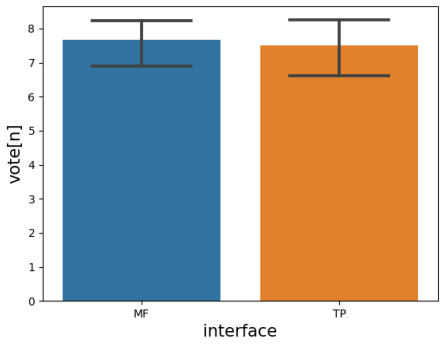


(g)执行时间（*ExT*）。

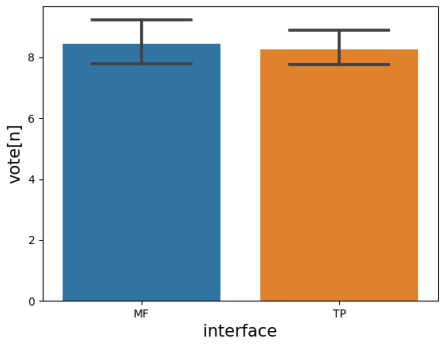


(h)界面易用性，表*1* 问题*2*。

图11. 任务*1* 实验结果。TP：UR10e 示教器接口。MF：操控框架界面。



(a)界面直观性，表*1 Q3*。



(b)界面速度，表*1 Q4*。

图12。任务*1* 实验结果。TP：UR10e示教器界面。MF：操作框架界面界面。

表2。任务1的参与者数据：E.Q.（教育资历）：h.s.g.（高中毕业）；m.d.（硕士学位）；b.d.（学士学位）。I.R.P.E.（工业机器人编程经验）。P.O.（职业）。

(a) A组

	Age	E.Q.	工业机器人编程经验	P.O.	性别
1	24	m.d.	少量	博士生	man
2	28	m.d.	no	博士生	man
3	35	高中毕业	少量	机械操作员	man
4	23	b.d.	no	学生	man
5	20	高中毕业	no	学生	女性
6	23	m.d.	少量	学生	man
7	26	m.d.	no	博士生	man
8	22	高中毕业	no	学生	man

（b）B组

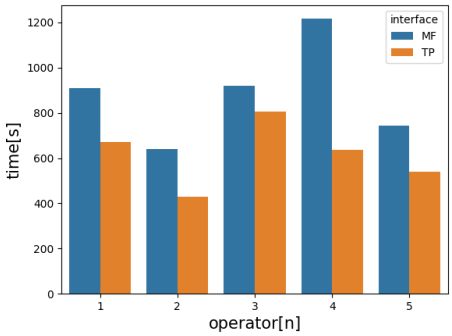
	Age	E.Q.	工业机器人编程经验	P.O.	性别
9	29	PhD	专家	研究人员	male
10	35	高中毕业	少量	机械操作员	male
11	23	b.d.	no	学生	女性
12	24	b.d.	少量	学生	male
13	24	b.d.	少量	学生	male
14	23	b.d.	no	学生	女性
15	22	高中毕业	no	学生	male
16	31	m.d.	no	软件开发人员	male
17	34	m.d.	no	软件工程师	male

4.2. 任务*2*实验结果

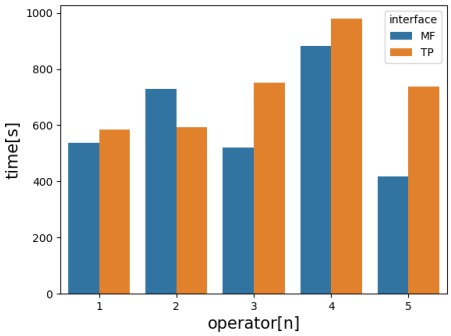
任务2相关实验采用了第3.3节和第3.3.2节所述的方法。第3.4节报告了用于评估界面的测量参数。任务2期间测得的所有评估指标均列于图13和图14中。

MFI的平均学习时间LeT高出43.7%；见图13a。该结果与任务1实验中的LeT相似。MFI的平均编程时间PrT低了15.2%；见图13b。两种界面的PrT相近，因为任务*2*仅包含两个对象，重复动作的数量减少，因此结果趋于相似。MFI的编程问题$PrQs$更多；见图13c。参与任务2的用户（即车间机床操作员和技术人员）对机器人使用的熟悉程度较低，且总体而言，相较于任务任务*1*的用户（即STEM院系学生），他们对技术的兴趣也较低。这可能是任务*2 PrQs*数值高于任务*1*的原因，尤其是在MFI中。MFI结构更复杂，进一步放大了这一现象。尽管存在这个问题，$PrQs$数值并不构成实际问题。对非专家操作员进行MFI培训所需时间不到一小时（包括观看视频）。测试时间TeT（见图13d）表明，两种界面的TeT相近；用户1的峰值是例外，但两种界面中均出现了这一情况；尤其是，纠正错误所需的时间相当。测试次数$TeNs$（见图13e）显示，试验次数并不多，因此编程过程中错误数量较少，这表明知识水平较低并不会影响操作员表现。测试问题$TeQs$（见图13f）表明，用户能够高度自主地纠正错误。执行时间ExT（见图13g）显示，TPI的ExT低于MFI。在这种情况下，操控框架生成无碰撞轨迹所需的计算时间比任务1更长，因为机器人工作空间中存在受限区域和更多障碍物。较高的$ExTs$体现了操控框架运动规划器所保证的无碰撞轨迹带来的益处。相反，使用TPI时，机器人碰撞规避由程序员负责。平均重新编程时间ReT（见图13h）表明，MFI的ReT通常低于TPI。MFI的ReT平均值比TPI低26.1%。使用MFI时，用户只需修改程序，示教两个新位置。相比之下，使用TPI则需要在已定义的轨迹中添加新的中间点。重新编程问题数$ReQs$（见图14a）显示$ReQs$数值较低；大多数用户在重新编程过程中不需要任何帮助。重新编程测试时间$ReTeTs$、重新编程测试次数$ReTeN$、测试问题$ReTeQs$和执行时间$ReExT$（分别见图14b、14c、14d和14e）所显示的结果与第一测试阶段所得结果相似，且界面之间未出现显著差异。

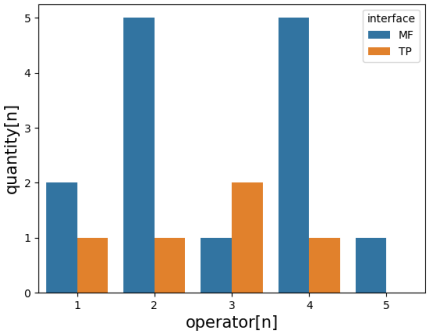
与任务1一样，在任务2结束时，向用户发放了表1所述的问卷。图14f、14g和14h再次显示，两种界面的数值相近。这些参数的得分较高，表明机床操作员对其评价良好，凸显了这些界面在工业领域的可用性。此前知识的评估未提供有价值的信息，因此未报告问题1和问题5的得分图。表3列出了参与任务2实验的用户信息。



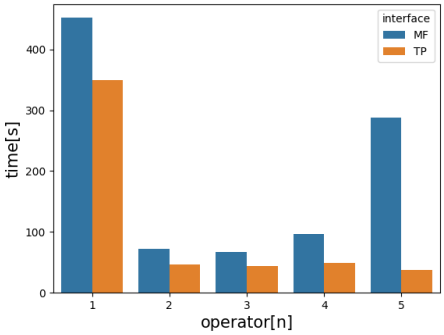
(a)学习时间（*LeT*）。



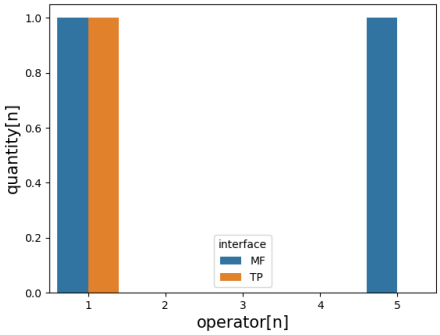
(b)编程时间（*PrT*）。



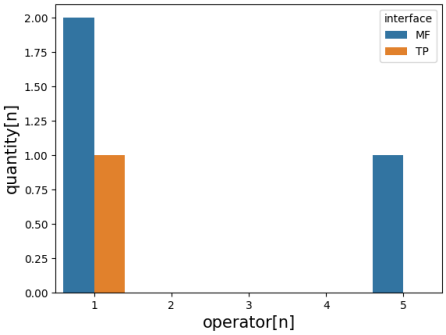
(c)编程问题（*PrQs*）。



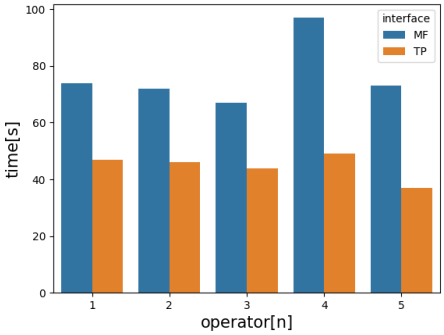
(d)测试时间（*TeT*）。



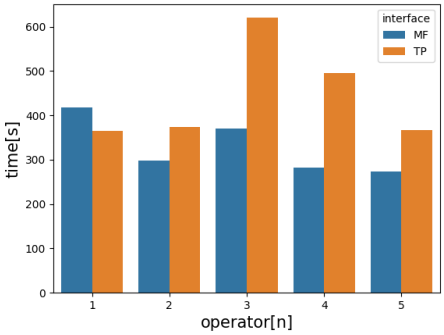
(e)测试数量（*TeNs*）。



(f)测试问题（*TeQs*）。

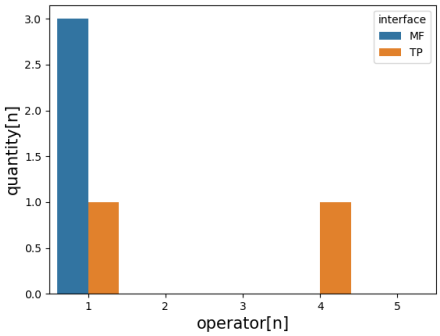


(g)执行时间（*ExT*）。

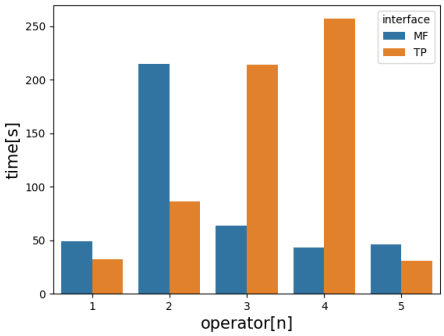


(h)重新编程时间（*ReT*）。

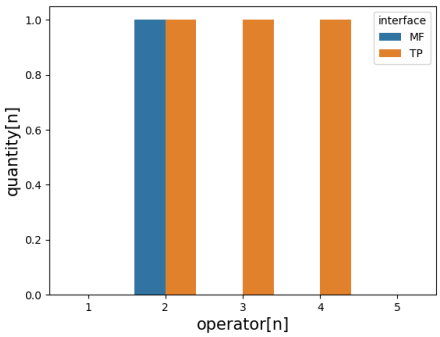
图13。任务*2* 实验结果。TP：UR10e 示教器。MF：操作框架界面。



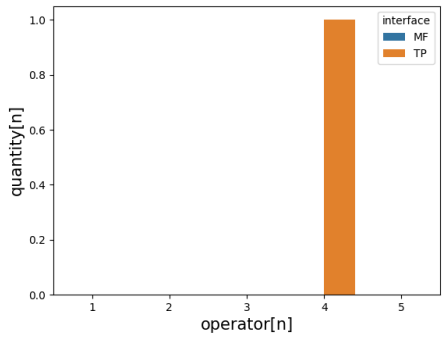
(a)重编程问题（*ReQs*）。



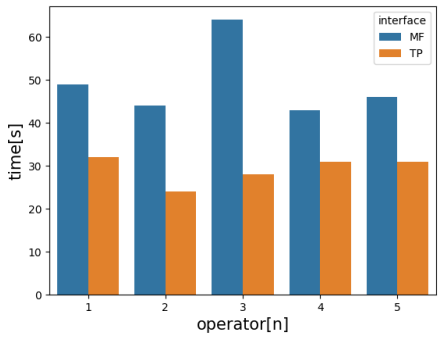
(b)重编程测试时间（*ReTeT*）。



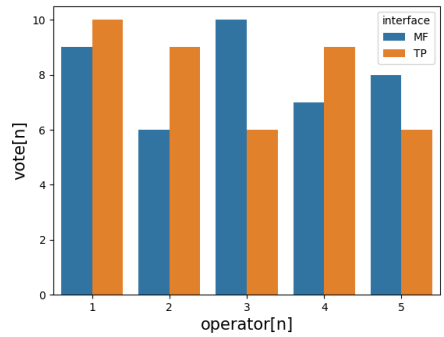
(c)重编程测试数量（*ReTeN*）。



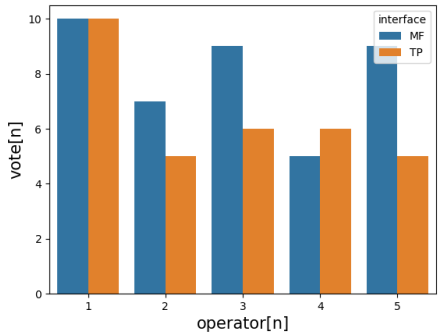
(d)重编程测试问题（*ReTeQ*）。



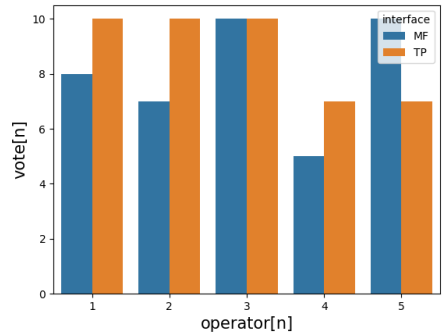
(e)重编程执行时间（*ReExT*）。



(f)界面易用性，表*1 Q2*。



(g)界面直观性，表*1 Q3*。



(h)界面速度，表*1 Q4*。

图14。任务*2* 实验结果。TP：UR10e 示教器。MF：操作框架界面。

表3。任务2的参与者数据：教育资历（E.Q.）：高中毕业（h.s.g.）；医学博士（m.d.）（硕士学位）；学士学位（b.d.）（学士学位）。工业机器人编程经验（I.R.P.E.）。职业（P.O.）。

	Age	E.Q.	工业机器人编程经验	P.O.	性别
1	36	高中毕业	少量	机械操作员	male
2	35	高中毕业	少量	机械操作员	male
3	32	高中毕业	No	机械操作员	male
4	46	高中毕业	No	轮班主管	male
5	26	m.d.	少量	工艺工程师	male

4.3. 讨论

两种界面的实验结果相似。界面在各个操作方面都表现出相似性。换言之，MFI 可以与 UR10e 示教器接口相媲美。当任务包含重复性动作时，通过日常使用操控框架进行机器人编程，可以迅速弥补 MFI 较长的学习时间。不过，非专家最终用户学习 MFI 所需的时间很短，不到一小时便能基本了解该界面，并能够独立进行任务编程。同时，操控框架还带来了多项额外优势，例如使用 ROS 框架、面向任务的编程以及无碰撞运动规划。本文通过将复杂的机器人编程框架与直观的界面相结合，实现了预期目标，证明车间操作员无需经历显著的学习障碍或具备特定的前期知识，便能迅速掌握该框架。本文并非旨在取代 UR10e 示教器，而是要证明，使用具有合适编程界面的复杂软件，可以让车间操作员也能使用高级编程功能，从而使中小企业能够采用先进的机器人单元。

5. 结论与未来工作

本文比较了两种代表不同机器人编程方法的机器人编程界面。第一种是 UR10e 示教器界面，结合引导式编程，实现了经典的面向机器人方法。第二种是操作框架界面，提供面向任务的编程方法。对 22 名用户开展的实验显示，两种界面的结果相近，这表明，直观的界面能够隐藏基于 ROS 的框架（如操控框架）的复杂性，从而在缺乏特定编程经验的最终用户中获得较高的认可度。较短的学习时间表明，只需很少时间就能培训最终用户，使车间无需具备特定机器人学知识，也能使用高级功能。同时，重新编程时的灵活性也有所提升。所有工厂操作员都可以修改机器人控制程序。

在未来的工作中，作者将研究如何利用机器学习算法创建新动作。例如，仅凭有限数量的技能，机器学习算法就应能根据具体要求动态组合出新动作，无需开发人员介入。

致谢

本研究部分得到了 ShareWork 项目（H2020，欧盟委员会——G.A. 820807）的支持。作者感谢 Cembre S.p.A.（Cembre S.p.A., 2022）提供机会，使他们能够在该公司的车间与机床操作员一起开展实验测试。特别感谢 Andrea Scala 和 Piervincenzo Tavormina 协助组织实验。

参考文献

Ajaykumar, Gopika, Maureen Steele 和 Chien-Ming Huang。2021年。“终端用户机器人编程综述。” *ACM Comput. Surv.*54 (8)。https://doi.org/ 10.1145/3466819。Ajaykumar, Gopika, Maia Stiber 和 Chien-Ming Huang。2021年。“为协作机器人的动觉示教设计以用户为中心的编程辅助工具。”《机器人与自主系统》 145：103845。https://www.sciencedirect.com/science/article/pii/S0921889021001305。Andronas, Dionisis, George Apostolopoulos, Nikos Fourtakas 和 Sotiris Makris。2021年。“用于自然人机交互的多模态界面。” *Procedia Manufac-turing* 54：197–202。第十届CIRP赞助数字企业技术会议（DET 2020）——数字技术作为提升工业竞争力和可持续性的推动力，https://www.sciencedirect.com/science/article/ pii/S2351978921001669。Anzalone, Salvatore M., Marie Avril, Hanan Salam 和 Mohamed Chetouani。2014年。“IMI2S：一种轻量级分布式计算框架。”收录于《自主机器人仿真、建模与编程》，Davide Brugali、Jan F. Broenink、Torsten Kroeger 和 Bruce A. MacDonald 编，Cham，267–278。施普林格国际出版。Apostolopoulos, George, Dionisis Andronas, Nikos Fourtakas 和 Sotiris Makris。2022年。“混合环境中的操作员培训框架：一种采用机器学习进行物体识别的增强现实模块。” *Procedia CIRP* 106：102–107。第九届CIRP装配技术与系统会议，https://www.sciencedirect.com/science/article/pii/S2212827122001639。Bascetta, Luca, Gianni Ferretti, Gianantonio Magnani 和 Paolo Rocco。2013年。“基于导纳控制的机器人操作臂漫游式编程。”*Robotica* 31 (7)：1143–1153。Billard, A., S. Calinon, R. Dillmann 和 S. Schaal。2008年。“示教编程。载于：《机器人手册》。”。Blankemeyer, Sebastian, Rolf Wiemann, Lukas Posniak, Christoph Pregizer 和 Annika Raatz。2018年。“利用增强现实进行直观机器人编程。”*Procedia CIRP* 76：155–160。第七届CIRP装配技术与系统会议（CATS 2018），https://www.sciencedirect.com/science/article/pii/ S2212827118300933。Bøgh, Simon, Oluf Skov Nielsen, Mikkel Rath Pedersen, Volker Kr¨uger 和 Ole Madsen。2012年。“你的机器人有技能吗？”收录于第*43*届国际机器人研讨会论文集，VDE Verlag GMBH。Castro, Andr´e, Jo˜ao Pedro Souza, Lu´ıs Rocha 和 Manuel F. Silva。2019年。“AdaptPack Studio：面向工厂环境的自动离线机器人编程框架。”收录于

ronments。”收录于*2019*年*IEEE*自主机器人系统与竞赛国际会议（*ICARSC*），1–6。Cembre S.p.A. 2022年。“Cembre S.p.A.” https://www.cembre.it/。Coronado, Enrique, Fulvio Mastrogiovanni, Bipin Indurkhya 和 Gentiane Venture。2020年。“面向智能社交机器人终端用户开发的可视化编程环境：系统综述。”*Journal of Computer Languages* 58：100970。https://www.sciencedirect.com/science/article/pii/S2590118420300307。Delledonne, Michele 和 Manuel Beschi。2022a。“操作界面。” https: // github.com/JRL-CARI-CNR-UNIBS/manipulation_interface。Delledonne, Michele 和 Manuel Beschi。2022b。“ 操作界面。” https: // github.com/JRL-CARI-CNR-UNIBS/manipulation_interface_example。Faroni, Marco, Manuel Beschi, Stefano Ghidini, Nicola Pedrocchi, Alessandro Umbrico, Andrea Orlandini 和 Amedeo Cesta。2020年。“面向人机协作的分层控制方法：兼顾人因的任务与运动规划。”收录于*2020*年第*29*届*IEEE*机器人与人机交互国际会议（*RO-MAN*），1204–1210。Ferraguti, Federica, Chiara TalignaniLandi, LorenzoSabattini, Marcello Bonf`e, Cesare Fantuzzi 和 Cristian Secchi。2019年。“用于稳定物理人机交互的可变导纳控制策略。”*The International Journal of RoboticsResearch* 38 (6)：747–765。https://doi.org/10.1177/0278364919840415。Gadre, Samir Yitzhak, Eric Rosen, Gary Chien, Elizabeth Phillips, Stefanie Tellex 和 George Konidaris。2019 年。“ 利用混合现实进行终端用户机器人编程。” 收录于 *2019* 年国际机器人与自动化会议（*ICRA*），2707–2713。Heimann, Oliver 和 Jan Guhl。2020年。“工业机器人编程方法：范围综述。”收录于*2020*年第*25*届*IEEE*新兴技术与工厂自动化国际会议（*ETFA*），第1卷，696–703。Huang, Gaoping, Pawan Rao, Meng-Han Wu, Xun Qian, Shimon Nof, Karthik Ramani 和 Alexander Quinn。2020年。“Vipo：面向机器人物联网工作流的函数式空间可视化编程。”收录于*CHI '20*：计算机系统中的人为因素会议，04，1–13。Huang, Justin 和 Maya Cakmak。2017年。“Code3：一个面向新手和专家、用于移动操作机器人端到端编程的系统。”收录于*2017*年第*12*届*ACM/IEEE*人机交互国际会议（*HRI*，453–462。国际机器人联合会。2022年。“《世界机器人：工业机器人报告》。” https://ifr.org/worldrobotics/。ISO/TC 299 机器人技术。2016。“ISO/TS 15066:2016。” https://www.iso.org/ standard/62996.html。Kubota, Alyssa, Emma I. C. Peterson, Vaishali Rajendren, Hadas Kress-Gazit 和 Laurel D. Riek。2020年。*JESSIE*：为个性化神经康复及其他领域合成社交机器人行为，121–130。美国纽约州纽约：美国计算机协会。https://doi.org/10.1145/3319502.3374836。Kuka AG。2022年。“Kuak Sunrise.OS。” https://www.kuka.com/en-gb/products/ robotics-systems/software/system-software/sunriseos。Lu, Shuang, Julia Berger 和 Johannes Schilp。2022 年。“ 从多模态示教和自然语言指令中学习的机器人系统。” *Procedia CIRP* 107：914–919。引领制造系统转型——2022年第55届CIRP制造系统会议论文集，https: //www.sciencedirect.com/science/article/pii/S2212827122003687。Makris, Sotiris, Panagiota Tsarouchi, Dragoljub Surdilovic 和 J¨org Kr¨uger。2014年。“面向装配作业的直观双臂机器人编程。”*CIRP An-*

nals 63 (1): 13–16。https://www.sciencedirect.com/science/article/pii/S0007850614000201。Metta, Giorgio、Paul Fitzpatrick 和 Lorenzo Natale。2006年。“YARP：又一个机器人平台。”《国际先进机器人系统杂志》3 (1): 8。https://doi.org/10.5772/5761。Michalos, George、Sotiris Makris、Jason Spiliotopoulos、Ioannis Misios、Panagiota Tsarouchi 和 George Chryssolouris。2014年。“ROBO-PARTNER：面向未来装配工厂智能、灵活且安全操作的无缝人机协作。”*Procedia CIRP* 23：71–76。第5届CATS 2014——国际生产工程学会装配技术与系统会议，https://www.sciencedirect.com/ science/article/pii/S2212827114011366。Mohammad Safeea、Pedro Neto。2022年。“利用手动引导实现协作机器人操作臂的精确定位。”《国际先进制造技术杂志》120：5497–5508。https://doi.org/10.1007/s00170-022-09107-1。Mukherjee, Debasmita、Kashish Gupta、Li Hsin Chang 和 Homayoun Najjaran。2022年。“工业环境中人机协作的机器人学习策略综述。”《机器人与计算机集成制造》73：102231。https://www.sciencedirect.com/science/article/pii/S0736584521001137。Neto, Pedro 和 Nuno Mendes。2013年。“通过通用CAD软件包直接进行离线机器人编程。”《机器人与自主系统》61 (8)：896–910。https: // www.sciencedirect.com/science/article/pii/S0921889013000419。Ong, S.K.、A.W.W. Yew、N.K. Thanigaivel 和 A.Y.C. Nee。2020年。“面向工业应用的增强现实辅助机器人编程系统。”《机器人与计算机集成制造》61：101820。https://www.sciencedirect. com/science/article/pii/S0736584519300250。开源社区。2022年。“MoveIt! 让机器人迈向未来。”https:// moveit.ros.org/。Pan, Zengxi、Joseph Polden、Nathan Larkin、Stephen Van Duin 和 John Nor- rish。2012年。“工业机器人编程方法的最新进展。”《机器人与计算机集成制造》28 (2)：87–94。https://www. sciencedirect.com/science/article/pii/S0736584511001001。Quintero, Camilo Perez、Sarah Li、Matthew KXJ Pan、Wesley P. Chan、H.F. Machiel Van der Loos 和 Elizabeth Croft。2018年。“增强现实中的增强轨迹机器人编程。”载于*2018*年*IEEE/RSJ*智能机器人与系统国际会议（*IROS*），1838–1844。Rendiniello, Angelo、Alberto Remus、Ines Sorrentino、Prajval Kumar Murali、Daniele Pucci、Marco Maggiali、Lorenzo Natale 等。2020年。“面向机器人工业应用的灵活软件架构。”载于*2020*年第*25*届*IEEE*新兴技术与工厂自动化国际会议（*ETFA*），第1卷，1273–1276。ROBO-PARTNER联盟。2013年11月—2017年4月。“ROBO-PARTNER项目：面向未来装配工厂智能、灵活且安全操作的无缝人机协作。”http://www.robo-partner.eu/。ROS-Industrial联盟。2022年。“ROS-Industrial。”https://rosindustrial.org/。Schmidt, Douglas C.、David L. Levine 和 Sumedh Mungee。1998年。“TAO实时对象请求代理的设计。”*Comput. Commun.* 21 (4)：294–324。https://doi.org/10.1016/S0140-3664(97)00165-5。Schou, Casper、Rasmus Skovgaard Andersen、Dimitrios Chrysostomou、Simon Bøgh 和 Ole Madsen。2018年。“工业环境中协作机器人的基于技能的指令。”《机器人与计算机集成制造》53：72–80。https:// www.sciencedirect.com/science/article/pii/S0736584516301910。

Sefidgar, Yasaman S.、Prerna Agarwal 和 Maya Cakmak。2017年。“情境化有形机器人编程。”载于*2017*年第*12*届*ACM/IEEE*人机交互国际会议论文集，HRI ’17，美国纽约州纽约市，473–482。美国计算机协会。https://doi.org/10.1145/2909824.3020240。Sharework联盟。2018年11月—2022年10月。“Sharework。”https: //sharework-project.eu/。Sherlock联盟。2018年10月—2022年9月a。“Sharlock——面向新型协作工作场所的无缝、安全、以人为中心的机器人应用。”https://www. sherlock-project.eu/home。Sherlock联盟。2018年10月—2022年9月b。“Sherlock——交付成果4.8 自主学习与编程方法——最终原型。”https://ec.europa.eu/research/participants/documents/downloadPublic?documentIds=080166e5e30cb5db&appId=PPGMS。Siciliano, Bruno、Lorenzo Sciavicco、Luigi Villani 和 Giuseppe Oriolo。2010年。“机器人学：建模、规划与控制。”。斯坦福人工智能实验室等。2022年。“机器人操作系统。”https://www.ros.org。Stenmark, Maj、Mathias Haaae 和 Elin Anna Topp。2017年。“安全工业机器人可复用技能的简化编程——原型与评估。”载于*2017*年第*12*届*ACM/IEEE*人机交互国际会议（*HRI*，463–472。Thormann, Christian 和 Alexander Winkler。2021年。“使用功能块和顺序功能图对轻型机器人进行编程。”载于*2021*年第*25*届自动化与机器人学方法和模型国际会议（*MMAR*），348–353。Tsarouchi, Panagiota、Sotiris Makris 和 George Chryssolouris。2016年。“人机交互综述及任务规划与编程方面的挑战。”《国际计算机集成制造杂志》29 (8)：916–931。https://doi.org/ 10.1080/0951192X.2015.1130251。Umbrico, Alessandro、Andrea Orlandini、Amedeo Cesta、Marco Faroni、Manuel Beschi、Nicola Pedrocchi、Andrea Scala 等。2022年。“面向个性化人机协作的高级人机协作单元设计。”《应用科学》12 (14)：6839。http://dx.doi.org/10.3390/app12146839。Universal Robots。2022年。“Universal Robots ROS驱动程序。”https://github.com/ UniversalRobots/Universal_Robots_ROS_Driver。Villagrossi, E.、N. Pedrocchi 和 M. Beschi。2021年。“通过基于动作和技能的操作框架简化机器人编程。”载于*2021*年第*26*届*IEEE*新兴技术与工厂自动化国际会议（*ETFA*），1–6。Villagrossi, Enrico 和 Manuel Beschi。2022年a。“操作示例。”https:// github.com/JRL-CARI-CNR-UNIBS/manipulation_examples。Villagrossi, Enrico 和 Manuel Beschi。2022年b。“操作库。”https: // github.com/JRL-CARI-CNR-UNIBS/manipulation。Villani, Valeria、Fabio Pini、Francesco Leali 和 Cristian Secchi。2018年。“工业环境中的人机协作综述：安全性、直观界面与应用。”《机电一体化》55：248–266。https://www.sciencedirect.com/science/ article/pii/S0957415818300321。Yang, Shuo、Xinjun Mao、Binbin Ge 和 Sen Yang。2015年。“机器人编程语言的路线图与挑战。”载于*2015*年*IEEE*系统、人类与控制论国际会议，328–333。

Zhang、Jiafan、Yue Wang 和 Rong Xiong。2016。《工业机器人示教编程》。收录于*2016*年国际先进机器人与机电一体化会议（*ICARM*），300–305。