LLM-ASSISTED VR-ENABLED INTUITIVE ROBOTIC CONTROL SYSTEMATIC APPROACH FOR HUMAN-CENTRIC SMART MANUFACTURING

KE WAN

MPhil

The Hong Kong Polytechnic University

2025

The Hong Kong Polytechnic University

Department of Industrial and Systems Engineering

LLM-Assisted VR-Enabled Intuitive Robotic Control Systematic Approach for Human-Centric Smart Manufacturing

Ke Wan

A thesis
submitted in partial fulfilment of the requirements
for the degree of

Master of Philosophy

December 2024

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

KE WAN

(Signed)

(Name of student)

Abstract

With the shift towards personalized manufacturing and the growing demand for high customization following the introduction of the Industry 5.0 concept, traditional predefined programming approaches have become inadequate for meeting the increasingly complex demands of modern product manufacturing processes. Preprogrammed robots, which rely on rigid and fixed action sequences, face significant challenges in complex and dynamic production environments. While these robots are designed to perform repetitive tasks with precision, they lack the flexibility to adapt to changing conditions or unexpected variations in the production process. Consequently, any deviation from the programmed task necessitates manual reprogramming, resulting in limited effectiveness and increased operational costs. To address these limitations, human-centric manufacturing has emerged as a solution, enabling seamless integration of human intelligence with the precision and efficiency of robots. Unlike traditional preprogrammed robots, human-centric manufacturing systems are highly adaptable, responding dynamically to the variability and unpredictability inherent in customized manufacturing environments.

In recent years, significant research has focused on human-centric manufacturing, with an emphasis on enhancing interaction efficiency and integrating artificial intelligence (AI) for decision-making. These studies have investigated how collaborative robots can serve as more efficient interaction platforms and adapt to dynamic environments. However, despite these advancements, several notable research gaps persist. For example, while AI has been incorporated into certain planning processes, the potential of large language models (LLMs) for comprehensive robot task planning remains underexplored. Furthermore, existing research often falls short in developing intuitive robot control systems, leading to high learning cost and diminished user experience and precision for human operators. To address these challenges, this thesis aims to propose solutions focusing on three critical aspects of human-centric manufacturing scenarios: perception, planning, and execution.

In the investigation of robot task planning, a multi-modal pre-trained LLM is leveraged to seamlessly translate high-level human instructions into actionable robot commands, enhancing the interaction between human operators and robotic systems (Chapter 3). This approach is structured across three integral layers: task decomposer, motion descriptor, and robot code generator. Each layer is meticulously designed with structured prompts, including detailed templates and specific rules to ensure the generation of precise and effective outputs by the LLM agent.

In addition, a VR-based robotic control system is developed to enhance intuitive control and immersive visual feedback in robotic manipulation for teleoperation tasks (Chapter 4). The system leverages immersive VR interfaces to provide operators with real-time feedback and control mechanisms over robot actions, enabling seamless interaction in complex environments. By integrating intuitive VR input methods, immersive visual perception approaches, as well as seamless data exchange between human operator and onsite robot manipulator, the teleoperation system allows for manipulation of objects in complex manufacturing scenarios, facilitating efficient task execution.

Keywords: Human-centric smart manufacturing; cyber-physical system; virtual reality; large language model.

Acknowledgement

The journey to this academic milestone was paved by the encouragement and assistance of many, to whom I owe profound gratitude.

First and foremost, I would like to express my heartfelt gratitude to my Chief Supervisor, Dr. Zheng Pai. Dr. Zheng's invitation to undertake this advanced academic endeavor came at a pivotal moment in my life, offering me both direction and purpose when I was uncertain about my path. His exceptional professionalism, profound expertise, and genuine passion for research have been an unwavering source of inspiration throughout my journey. Dr. Zheng's thoughtful mentorship, constructive feedback, and ability to illuminate the broader academic context have been invaluable in shaping both my research and my academic growth. During times of doubt and difficulty, his steady encouragement and insightful advice consistently reignited my determination and reminded me of the greater vision behind my work. Working under Dr. Zheng's guidance has been an extraordinary privilege and a profoundly enriching experience for which I am deeply grateful.

I would like to extend my heartfelt thanks to all the members of the Research Group of AI for Industrial Digital Servitization (RAIDS). Their unwavering

support, insightful discussions, and collaborative spirit have been integral to my research journey. The stimulating academic environment they foster, combined with their encouragement and camaraderie, has not only enriched my research experience but also made this journey deeply rewarding. I am truly grateful for their contributions, both professionally and personally, during my time with the group.

My deepest love and gratitude go to my family for their unwavering understanding and unconditional support. Their constant encouragement and belief in me have been a source of strength and motivation throughout this journey.

Contents

1 Introduction **1**

1.1 Background 2
1.2 Research Scope 5
1.3 Research Objectives 7
1.4 Thesis Structure 8

2 Literature Review **10**

2.1 LLMs and Robot Task Planning 10
2.1.1 Traditional Approaches to Robot Task Planning 13
2.1.2 LLMs in Robot Task Planning 16
2.2 VR and Robotic Control 18
2.2.1 Robotic Control Definition 20
2.2.2 VR Definition 22
2.2.3 Robotic Control Systems 23
2.3 Research Gaps 31
2.3.1 LLMs for Robot Task Planning 31
2.3.2 VR-based Intuitive Robotic Control 32

3 LLM-based Robot Task Planning **34**

3.1 Introduction 34
3.2 LLM-based Multi-Layer Robot Task Decomposition Planning . 37
3.2.1 Task Decomposition 39
3.2.2 Motion Description 40
3.2.3 Robot Code Generation 41
3.3 Experimental Results 44
3.4 Chapter Summary 46

4 Intuitive Robot Control for Complex Environments **48**

4.1 Introduction 49
4.2 VR-based Robot Teleoperation System 52
4.2.1 Overall System Design and Implementation 53
4.2.2 Interaction Module 55

4.2.3 Motion Planning Module 67
4.3 User Study 78
4.3.1 Experimental Setup 79
4.3.2 Results and Analysis 80
4.3.3 Discussion 83
4.4 Chapter Summary 85

5 Conclusions **87**
5.1 Contributions 88
5.2 Limitations 89
5.3 Future Research Directions 91

References **94**

List of Figures

1.1 Research scope and content organization of this thesis. 6

3.1 LLM-based and human-prompt guided task planning and robot code generation. 38
3.2 Gear pump components used in the assembly task. 40
3.3 Example executable code generated by code generator. 44

4.1 An operator teleoperating a robot via the proposed system. The work area information is captured and visualized in the virtual environment, where a virtual representation of the robot arm reflects the real robot pose. 53
4.2 Framework of the proposed VR teleoperation system. 54
4.3 Point cloud data collected by RGB-D camera before and after calibration: (a) before calibration; (b) after calibration. 60
4.4 Overview of the framework of our Unity-ROS communication established based on TCP connections [113]. 61
4.5 Joint definitions of the human hand. 65
4.6 Fundamental node-based framework of ROS communication. [114] 68
4.7 Overview of MoveIt's Move Group interface. [115] 70
4.8 RRT-Connect planning algorithm [119]. 72
4.9 Proposed robot motion planning module in the teleoperation system. 73
4.10 Human and robot hand motion mapping. 78
4.11 Representative sample results of the user study. (a) A participant is operating using VR equipment. (b) The Unity virtual environment interface during the operation process, with the bottom section showing the operator's perspective in VR. (c) Grasping components during the gear pump assembly operation. 81
4.12 Questionaire results (5-point Likert scale, average in parentheses). 81

List of Tables

3.1 Robot primitive skills and corresponding APIs. 43
3.2 Results of LLM-based task planning. 45

4.1 Command frame format for writing to the robot hand's registers. 75

1

Introduction

The future of manufacturing is shifting towards a human-centric paradigm, where humans and intelligent systems collaborate seamlessly to achieve high levels of flexibility and efficiency [1]. As the demand for large-scale personalized products grows and customization becomes a central requirement in today's manufacturing industry, traditional rigid automation approaches are becoming inadequate [2]. This shift towards mass personalization has catalyzed the emergence of human-centric manufacturing as a critical enabler, integrating the adaptability and decision-making capabilities of humans with the precision and high efficiency of robots. In this context, intuitive robotic control systems, enhanced by advanced technologies such as virtual-reality (VR) and large language model (LLM), are critical to overcoming challenges in perception, planning, and execution processes during human-centric manufacturing [3]. In this chapter, we begin by exploring the background and motivation of this research, emphasizing the role of human-centric manufacturing in the modern smart manufacturing industry. Then, the scope and objectives of this research are given. Finally, the structure of the thesis is shown at the end of this chapter.

1.1 Background

In the context of Industry 5.0, the manufacturing industry is shifting from traditional mass production to mass customization to meet the growing demand for personalized product requirements [4]. This transition offers significant benefits, including increased flexibility, enhanced customer satisfaction, and a competitive edge in rapidly changing markets. Meanwhile, industrial robots are widely utilized across various stages of production to enhance efficiency [5]. However, this shift presents considerable challenges for preprogrammed robots designed for traditional automated manufacturing. Traditional preprogrammed robots, which excel in repetitive and precision-required tasks, fall short in dynamic and complex production environments due to their reliance on static programming. They lack the adaptability to respond to changing processes or product variations, often requiring costly and time-consuming reprogramming. To address these limitations, human-centric manufacturing has emerged as a promising solution. New industrial robot and collaborative robot programming approaches are required. The first generation of collaborative robot programming methods, initially marketed as flexible and intuitive, often failed to support the dynamic requirements of human-centric manufacturing. As a result, the industry is moving toward more advanced programming solutions that leverage technologies such as large language models and artificial intelligence code assistants. This paradigm places humans at an essential factor during the production process, fostering

combination and collaboration between human creativity and robotic precision [1]. Humans contribute to adaptability and decision-making capabilities, while robots handle precision-dependent tasks. By integrating their strengths, human-centric manufacturing enhances flexibility, improves efficiency, and provides the adaptability needed for modern personalized manufacturing systems [6].

In human-centric manufacturing systems, improving the efficiency of tasks collaboratively performed by humans and robots requires careful consideration of several key factors, including planning, perception, and execution [7]. In previous production systems, industrial robots typically lack the ability to perceive and plan, relying heavily on predefined instructions and static programming. At the same time, humans are unable to effectively assist robots in execution, as robotic systems are designed to operate independently and repetitively. This disconnect often led to inefficiencies, particularly in complex or dynamic production scenarios. Many research studies have been conducted in addressing these challenges, exploring ways to enhance robot perception, integrate advanced task planning methods, and enable more seamless human-robot interaction during execution [8]. Nevertheless, notable problems remain, such as the lack of intuitive control mechanisms and insufficient integration of task planning technologies into manufacturing workflows.

In recent years, LLMs have been introduced into task planning within the field of manufacturing, offering new possibilities to improve the flexibility and efficiency of robotic systems [9]. Existing studies have been focused on using LLMs for high-level planning tasks, such as breaking down complex goals into subtasks or generating structured task descriptions. These studies highlight the ability of LLMs to process vast amounts of textual data and provide logical task sequences, contributing to better decision-making in manufacturing systems. However, these studies have significant limitations when it comes to enable robots to directly execute actions, as they struggle to generate low-level robot commands, particularly in scenarios where both natural language instructions and visual inputs, such as images, need to be processed simultaneously. Many existing research studies do not focus on how LLMs can bridge the gap between human-provided instructions and the direct execution of robotic actions. This limits the practical applications of current LLM-based task planning in manufacturing environments where efficiency and precision are in demand.

On the other hand, VR has been increasingly introduced recently to enhance perception and execution capabilities within current human-centric manufacturing systems [10]. VR offers an immersive and interactive interface, allowing operators to visualize and control robotic systems onsite. This immersive experience provides significant advantages, such as improving spatial awareness, enabling intuitive manipulation in complex environments, and facilitating more dynamic interactions between humans and robots. Despite

its potential, existing VR-related research in manufacturing has notable limitations. Many current systems lack sufficiently intuitive control methods, relying on complex input mechanisms that significantly increase the learning cost for operators. Furthermore, current VR implementations often fail to incorporate reliable visual environmental perception capabilities, limiting the system's ability to provide real-time feedback for human operators.

To address the challenges mentioned above, this study aims at fully harnessing the potential of LLMs and VR technologies to enable a more seamless integration of human and robotic capabilities. Therefore, in this thesis, we propose a systematic approach for human-centric manufacturing scenarios, leveraging capabilities of humans and robots based on LLM and VR. Specifically, the LLM will handle task planning and generate executable code for robots, while VR will serve as the foundation of the robot control system, providing a more intuitive human-robot interaction interface for human operators.

1.2 Research Scope

This research focuses on addressing key challenges in human-centric manufacturing: making human-robot working environment more flexible and intuitive for human operators, by leveraging the capabilities of LLMs and VR technologies. The primary goal is to enhance the integration of human and robotic systems in dynamic and complex production environments, improving

efficiency and adaptability. Specifically, this research explores three critical aspects: (1)planning, where robot task are planned in details before action execution;(2)perception, where human operators are aware of the working environment on the robot side; and (3)execution, where robot manipulator executes human commands.

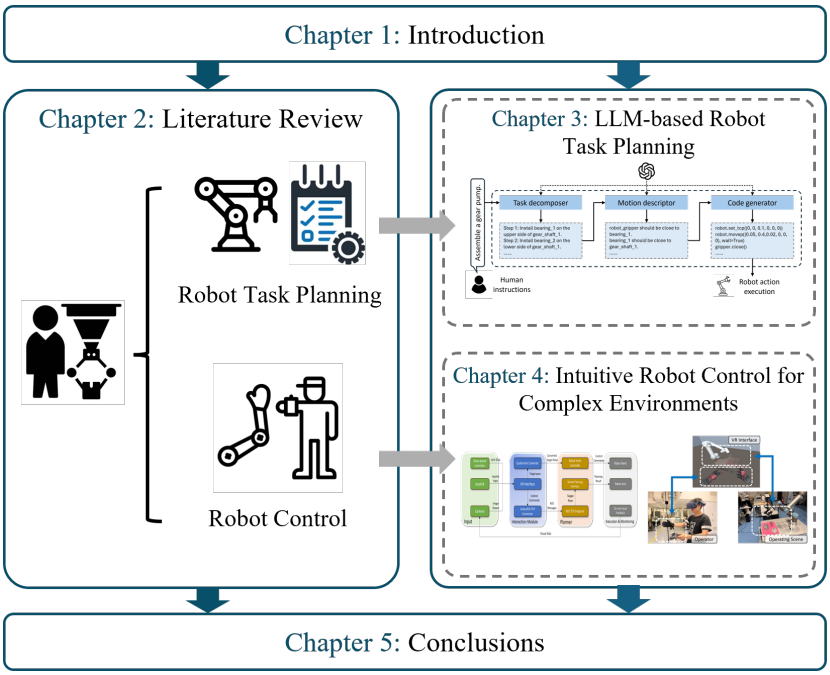

Fig. 1.1: Research scope and content organization of this thesis.

In the area of planning, this research explores the application of LLMs in robotic task planning. This includes developing a systematic framework that translates high-level human instructions, expressed in natural language and supported by visual inputs, into low-level executable robot commands. This work focuses on multi-modal inputs and structured task decomposition to improve task clarity and reduce reliance on manual programming.

For perception and execution, this study integrates VR technology to create an intuitive and immersive control interface for robotic systems. A com-

prehensive framework is discovered for VR-based robot control to ensure smooth information flow and data exchange. This framework enables real-time interaction and seamless communication between humans and robots. Within the robot control framework, the scope also includes designing and developing a VR interaction interface that reduces the learning cost for operators by providing user-friendly interaction methods. Additionally, the VR system incorporates real-time environmental perception leveraged by 3D visual feedback mechanisms . The research scope and content organization are illustrated in Fig. 1.1.

1.3 Research Objectives

The primary objective of this research is to propose an LLM-assisted VR-enabled intuitive robotic control systematic approach to address critical challenges in human-centric smart manufacturing. This approach is introduced to overcome limitations in planning, perception, and execution. Specifically, detailed research objectives of this thesis are stated as follows.

Objective 1: To develop a multi-layer LLM-based robot task planning method that converts human natural language instructions into executable robot commands.

The first objective is to design a robot task planning framework that uses LLMs to translate high-level human instructions into low-level executable robot

codes. This task planner incorporates natural language and visual inputs, enabling seamless communication between humans and robotic systems. By introducing multi-layer structured prompts, including a task decomposer, a motion descriptor, and a robot code generator, this framework ensures accurate and efficient task decomposition planning, as well as enables direct robot action execution.

Objective 2: To propose a VR-based robotic control system that enables intuitive and seamless robotic control and comprehensive visual awareness.

The second objective is to create a VR-enabled robotic control system that improves operator interaction and enhances user experience. This system leverages a module-based data exchange mechanism developed integrated with a virtual environment to ensure real-time communication between human operators and robots, enabling seamless robot control and perception. The VR interface reduces the learning cost by offering an intuitive and immersive robot control environment, while also incorporating real-time 3D visual feedback to provide comprehensive scene perception and understanding.

1.4 Thesis Structure

The remainder of the content in this thesis is organized as follows.

Chapter 2 reviews previous works related to the implementation of LLM and VR technologies modern manufacturing systems. While highlighting their advancements and contributions, limitations and challenges will be discussed to determine the existing research gaps.

Chapter 3 presents the development of the LLM-based task planning method. It details the proposed multi-modal approach for translating high-level human instructions into low-level executable robot codes, including the use of structured prompts, task decomposition, motion description, and code generation. Experimental results are discussed to validate the method's effectiveness.

Chapter 4 introduces our proposed VR-based robotic control system. This chapter provides a detailed explanation of the system's framework and modular components, including the virtual environment interaction, robot planning module, and data exchange mechanisms. It also discusses how these methods are employed to enable bidirectional information exchange between humans and robots, ultimately delivering a robot control and monitoring interface based on a virtual environment. A user study has been conducted to evaluate the system's feasibility and efficiency, and potential future development directions are discussed.

Lastly, Chapter 5 summarizes the main research contributions and limitations of this research study, as well as discussing future directions of this project.

2 Literature Review

In this chapter, a comprehensive literature review is presented to provide an exploration of the foundational elements relevant to LLM robot task planning, as well as VR-enabled robotic control in the context of human-centric smart manufacturing. It examines existing research on LLM-based robotic task planning, focusing on methodologies for task reasoning, adaptive decision making, and interaction design to enhance robotic autonomy and efficiency. Additionally, this chapter delves into VR-based robotic control, analyzing approaches for immersive interaction, real-time feedback, and intuitive operation within complex industrial manufacturing scenarios. This review also identifies critical gaps and emerging opportunities, offering a cohesive understanding of the current state of the field. This analysis not only highlights the challenges but also lays the groundwork for advancing human-centric and intelligent robotic control systems.

2.1 LLMs and Robot Task Planning

In the era of smart manufacturing, robotic systems in industrial manufacturing environments have been transitioning from traditional structured frameworks into unstructured human-centric paradigms [11]. These systems are expected to become indispensable for enhancing productivity, precision,

and flexibility in dynamic and complex production environments. Unlike traditional manufacturing systems that rely on rigid automation, human-centric manufacturing emphasizes the seamless integration of human expertise and robotic capabilities to handle diverse and customized tasks [10]. This shift has driven the development of more adaptive and intelligent robotic systems capable of understanding and executing high-level human instructions.

Central to this capability is robot task planning, which serves as the bridge between human operators and robotic execution. Task planning involves translating abstract, high-level instructions into structured, executable actions, enabling robots to act autonomously in real-world scenarios [12, 13]. As manufacturing processes become increasingly complex and dynamic, task planning has emerged as a critical component for ensuring the adaptability and scalability of robotic systems.

Traditional approaches to task planning can be broadly categorized into rule-based methods and learning-based methods, both of which have been extensively studied and applied in robotic systems [13]. Rule-based methods, such as STRIPS and PDDL, rely on predefined logical representations of the task environment and the use of symbolic reasoning to generate action sequences [14, 15, 16, 17]. These approaches excel in structured and predictable environments where the rules and constraints are explicitly defined. However, their reliance on manually crafted rules and domain models makes them inflexible and unsuitable for dynamic or unstructured environ-

ments, where task requirements or environmental conditions may change unpredictably. In contrast, learning-based methods leverage data-driven implementation, such as deep learning and reinforcement learning (RL), to train robots to perform tasks by learning from large datasets or through trial-and-error interactions with the environment [18, 19, 20]. These methods have demonstrated significant potential in handling complex tasks, such as robotic manipulation and motion planning, especially in scenarios where explicit domain knowledge is difficult to encode. Despite their adaptability, learning-based approaches face challenges such as high training costs, the need for large-scale labeled datasets, and difficulties in generalizing to unseen tasks or environments.

Recently, the emergence of LLMs, such as GPT-3 [[21], GPT-4 [22], and BERT [23], has revolutionized natural language processing (NLP) through their powerful capabilities in understanding, generating, and reasoning with human language. These models have demonstrated exceptional performance across various domains, including text processing [24], translation [25], and code generation [26]. In the context of robotics, LLMs introduce a paradigm shift by enabling robots to process unstructured natural language instructions and generate logical task plans without extensive manual programming [9]. This capability bridges the gap between human operators and robots, fostering more intuitive communication and task execution.

Despite their promise, research studies delved into LLM robot task planning are still limited. In this section, we review the state-of-the-art in robot task planning, highlighting the advancements, limitations, and potential research directions in this emerging field.

2.1.1 Traditional Approaches to Robot Task Planning

Robot task planning has long been a cornerstone of robotics research, enabling robots to autonomously generate and execute action sequences that achieve the goals. In recent decades, many research studies have been conducted on this topic. Typically, traditional approaches to task planning can be broadly categorized as classical task planning methods and learning-based methods, each offering distinct strengths and limitations [13].

Classical task planning focuses on deterministic and fully observable environments. Tasks are often modeled as state-transition problems, where the objective is to determine a sequence of actions that transition the system from an initial state to an expected goal state [27]. These methods rely on predefined models of the environment and logical reasoning, which makes them particularly effective in structured and predictable domains. State-space search is one of the foundational methods in this category [28, 29]. It relies on heuristics to guide the exploration of the state space, improving the efficiency of the search. For instance, relaxed planning graphs [30, 31],

which ignore the negative effects of actions, often serve as a basis for heuristic computation. These graphs simplify the planning problem, allowing for computationally efficient estimation of the minimum number of steps needed to reach the goal state. Despite their effectiveness in structured environments, state-space search methods typically struggle with scalability and adaptability in dynamic settings.

Another classical approach, hierarchical task network (HTN) planning, decomposes high-level tasks into smaller, more manageable sub-tasks [32, 33]. This hierarchical decomposition leverages domain-specific knowledge, allowing planners to focus on abstract task representations before refining them into concrete action sequences. While HTN planning is particularly effective in domains where tasks can be naturally structured into hierarchies, it faces significant challenges when applied to robotics. Specifically, its reliance on predefined task structures makes it difficult to adapt to dynamic or unpredictable scenarios, and its integration with low-level motion planning remains a complex issue. Temporal logic-based planning represents yet another classical approach, extending traditional methods to handle time-dependent tasks through formal frameworks like linear temporal logic (LTL) [34]. These methods often employ automata-based techniques to synthesize controllers that guarantee task completion under specified temporal constraints. While theoretically robust, their computational complexity limits their practical application in robotics, especially in environments with significant uncertainty.

In contrast to classical approaches, learning-based methods aim to address their limitations by leveraging data-driven techniques to improve scalability and adaptability [13]. By learning task representations, action effects, and planning strategies directly from data, these methods reduce reliance on manually crafted models and enable robots to operate in more dynamic and uncertain environments. One early direction in this area involved learning symbolic representations for planning. For example, in [35],probabilistic relational models were developed to represent action effects in uncertain environments. Similarly, in [36], deterministic models were applied to partially observable domains.

Recent advancements in deep learning have further expanded the capabilities of learning-based task planning [37]. Neural networks have been used to learn task representations from large-scale datasets, allowing planners to generalize across a wide variety of tasks. For instance, neural task programming decomposes task demonstrations into executable primitives, while neural task graphs represent tasks as compositional structures with nodes corresponding to actions and edges capturing dependencies [38, 39]. These approaches are particularly effective in handling long-horizon tasks, where sequential dependencies play a significant role. RL has also been widely applied to task planning, enabling robots to learn policies through trial-and-error interactions with their environment [40]. Deep RL models, for example, have been used to predict intermediate sub-goals or discrete actions, aiding in the efficient exploration of the action space [41]. Moreover, techniques

such as affordance-based learning allow robots to anticipate the long-term effects of their actions, improving their ability to achieve complex objectives in dynamic settings [42].

Despite their progress, these paradigms face limitations that motivate the development of novel approaches, such as LLMs, which aim to combine the strengths of classical reasoning with the adaptability of data-driven techniques.

2.1.2 LLMs in Robot Task Planning

In recent years, the emergence of LLMs, such as ChatGPT [43], PaLM [44], and Gemini [45], has revolutionized artificial intelligence by achieving remarkable performance across diverse tasks. These transformer-based models, trained on massive datasets, excel in NLP, demonstrating capabilities like contextual understanding, reasoning, and instruction following [46, 47, 48]. Their versatility extends to other fields, with multimodal models incorporating both language and vision pushing boundaries further [49, 50]. This rapid progress has reshaped AI research paradigms and highlighted the potential of LLMs in achieving general intelligence.

Due to its capabilities, LLMs have begun to play a transformative role in robotics by bridging the gap between natural language and scene understanding and robotic control. LLMs excel at interpreting ambiguous commands,

extracting implicit information, and resolving contextual uncertainties, enabling robots to better navigate and interact in dynamic environments [51]. Besides, vision-language models have further expanded robotic capabilities, allowing systems to combine visual understanding with linguistic input for tasks like command execution [52, 53]. Additionally, LLMs play a vital role in human-robot interaction by facilitating adaptive learning and personalized responses, improving robots' capacity to align with user preferences and behaviors [54].

Among these applications, LLMs are particularly well-suited for robot task planning due to their ability to interpret high-level natural language instructions, reason over complex contexts, and generate structured outputs. They can parse ambiguous or incomplete commands, infer implicit goals, and decompose tasks into actionable steps. Their vast pre-trained knowledge enables robots to adapt to diverse scenarios without requiring extensive task-specific programming. In recent years, significant efforts have been devoted to exploring this field. For example, in [55], the authors proposed leveraging LLMs for zero-shot task planning, where natural language instructions are transformed into high-level action plans. Similarly, [56] demonstrated how LLMs can infer context-based action sequences, such as deducing cleaning steps and tool usage in a "tidy the desk" task. In [57], they integrated language understanding with visual perception, enabling robots to identify target objects and plan actions in complex settings, such as locating and moving specific items. Furthermore, approaches like [58] utilize LLMs to au-

tomatically generate hierarchical task plans, progressively refining high-level goals into actionable sub-tasks.

These studies highlight the potential of LLMs to enhance robot task planning, offering significant advancements toward more adaptable and intelligent robotic systems. Nevertheless, several limitations persist. Most existing methods struggle to effectively generate low-level robot-executable commands, often producing generalized plans or abstract descriptions that require additional processing or manual intervention. In addition, many approaches lack a structured framework to handle the hierarchical nature of task planning, resulting in fragmented or inconsistent outputs. Furthermore, while LLM-based methods have shown potential in general robotics tasks, their application in specific domains like manufacturing, production assembly, and other industrial contexts remains limited, which demand tailored solutions that can handle domain-specific constraints.

2.2 VR and Robotic Control

Ever since first introduced in addressing nuclear waste management issues in 1954 [59], robot control systems have undergone decades of development. Numerous robot control systems have been proposed and designed to perform specific operational tasks [60]. Due to the challenges in achieving fully autonomous task completion, robot systems that involve human intervention for operation are a more practical choice [61]. In recent years, with the

introduction of the Industry 5.0 concept and the continuous development of human-robot collaboration technologies, the human-centered industrial production paradigm has gained significant attention. This has led to a surge in research related to robot control [5]. Current research on control systems primarily focuses on several aspects, including the development of user-friendly human-robot interaction interfaces, robotics learning algorithms, and the design of control architectures and learning strategies [60].

Meanwhile, Virtual Reality (VR) has brought significant promotions to industry development as it provides a natural and intuitive human-robot interaction interface. VR technology aims to provide users with an immersive virtual space and create interactive experiences that closely resemble real environments through various sensors and feedback devices. With the continuous enhancement of 3D engines such as Unity and Unreal, as well as the introduction of machine learning, computer vision, and other state-of-the-art technologies, VR has evolved beyond simply providing a virtual environment and has extended to the realm of replicating real environments and reconstructing them within the virtual realm [62]. Furthermore, in recent years, with the continuous maturation of hardware technology, the entry barriers for VR devices are gradually decreasing. This has led to an increasing number of researchers exploring the integration of VR technology into specific fields, such as remote control of robotic systems [63].

One of the goals of robot control is to create an intuitive interaction interface for human operators and provide them with as much environmental information around the robot as possible, and thus to enable operators to control the robot more naturally and seamlessly. Since VR can provide environmental information via 3D reconstruction, it is natural that VR is integrated into a robot control system as an interaction interface. Especially, under the wave of human-centric concepts such as Industry 5.0 and Society 5.0 [64, 65], VR and robot control can bring significant promotion to the human-robot relationship as enabling technologies. VR has the potential to bring about possibilities for HRC-based robotic control.

2.2.1 Robotic Control Definition

Robotic control typically refers to remote operation or remote manipulation, often involving vehicles or mechanical systems [66]. In this study, robot control is exclusively limited to robots as the subject. Robot robot control is a means to operate or collaborate with a robot utilizing human intelligence, which necessitates a human-machine interaction interface capable of providing adequate and comprehensive information [67]. To clarify this concept, the applications of robot control are often classified into three classes [68]:

- Closed loop control. The human operator controls the actuators of the robot via direct signals.

- Coordinated teleoperation. Similar to closed loop control, but this time some internal control loops are involved within the robot system.

- Supervisory control. The robot can perform part of the tasks more or less autonomously. The human operator mainly monitors and provides high-level control commands.

Another related concept is robot control. This introduces the possibility of providing a person with the feeling of actually being present at a remote location. In the field of robot control, telepresence typically refers to remotely controlled robot system that involves comprehensive information such as visual and haptic feedback, enabled by state-of-the-art technologies including computer graphics and VR [69]. Different from the Master-Slave teleoperation mode which involves a local robot (master) and a remote robot (slave), telepresence systems often combine a capture system, a network transmission system and a display system which establishes an interaction interface for the human operator to obtain comprehensive perception of the remote scene [67].

In this research, the definition of robotic control will incorporate elements of telepresence. Unlike traditional Master-Slave teleoperation, the concept discussed here primarily focuses on the interaction interface between human operators and controlled robots. operators rely on the perceptional information provided by the interface to understand the environmental conditions.

The robots are not fully autonomously controlled, while human operators require a certain level of control over the robots.

2.2.2 VR Definition

VR is an immersive technology that simulates a computer-generated environment, which can be experienced through specialized devices such as head-mounted devices (HMDs) [70]. By creating a sense of presence and interaction, VR transports users to artificial worlds that can mimic real-life environments. Furthermore, users are not mere observers but active participants within the virtual environment enabled via various input devices. They can explore and interact with objects and elements in the digital realm through gestures, motion tracking or handheld controllers. The immersive nature of VR enables users to perceive depth, scale and spatial relationships, enhancing the sense of interaction and engagement.

Command VR devices consist of VR HMDs, joysticks, data gloves, motion trackers and base stations. These devices are responsible for gathering input commands or motion information from users, while also displaying virtual scenes to provide an immersive interactive environment. Virtual scenes are typically provided by 3D engines and rely on hardware computation for real-time rendering. VR HMDs are typically divided into two categories: standalone VR and PCVR. Standalone VR relies on onboard computational power and is generally limited to running pre-packaged applications, while

PCVR depends on an external PC and enables real-time communication with the PC. In telerobotic systems, PCVR is predominantly utilized in research as it provides better performance and real-time capabilities [71].

2.2.3 Robotic Control Systems

Establishing a connection between human operators and robots forms the foundation of our VR-based robot control system. In this section, related research studies focused on information exchange between human operators and robots are reviewed. The main content can be divided into 2 aspects: traditional robot control methods, and visualization for telepresence awareness.

1) Robot Control Interface

The remote control of robots by human operators has been studied for years [72]. Different from plain robotic systems, where the robot executes a motion or other predefined program without further consultation or human users, robot control systems provide information to and require control commands from the human operator. In this section, we discuss the user command input methods proposed in existing research on robot control systems.

Many existing research studies implemented traditional desktop interface for remote robot control. Desktop interface typically involves 2D graphical user interface (GUI) and button-based command input devices, such as keyboards

and mouses and other similar devices which are familiar to most users. For example, [73] proposed a server-client-based teleoperation interface for continuum robot control. The server loads the user interface on a webpage-based GUI, and the operator clicks the mouse buttons and drags the 3D model displayed on the GUI to change the shape parameters of the robot. Meanwhile, a feedback segment integrated within the GUI allows the user to see the length, curvature and orientation for a given section. [74] presented a design for a high-level robot control interface that includes error handling and supports control over the failure recovery action. The user interface (UI) was developed based on a 2D desktop monitor interface for high-level control where the multiple views captured from RGB cameras, as well as some semantic information feedback and action suggestion are displayed, and a gamepad interface for low-level robot control. In [75], authors proposed a cable-driven parallel robot control system based on a master-slave framework. The operator controls the remote robot using a joystick and a trackball together and the 3D model of the remote robot is visualized and displayed on a desktop monitor. [76] designed an infrared-matrix-based robot control platform where the user controls a multi-DoF robot's joint pose via a touch screen integrated with infrared sensors. User's touch points are detected by the sensory system, and the visualization of the robot is achieved via 3D models.

Although 2D desktop interfaces are easy to use, they may not be able to reflect the natural way that humans observe, perceive and interact with the

real environment [71]. In recent years, growing efforts have been made to explore more intuitive methods to control robots. These research studies typically consider focusing attention on the ergonomic factors of input methods. For example, [77] introduced a robot control method based on depth image data, which interprets human hand gestures captured by a camera, removing the reliance on specialized controllers. This approach enables users to guide robots through natural hand movements, replicating the simplicity of manipulating objects in daily life. In [78], authors utilize a gamepad to set waypoints for the virtual surrogate of the drone based on an augmented reality (AR) interface. [79] developed an intuitive robot control system for controlling a 6-DoF industrial robotic manipulator using a Geomagic Touch haptic interface. The system integrates both virtual and physical sensor-based haptic feedback, enhancing the operator's environmental awareness and ensuring safer robot operation. [80] provided a cost-effective robot control system that utilizes Leap Motion interface to enable users to control the robot manipulator using their hand gestures. In [61], authors proposed a bilateral teleoperation system for wheeled mobile robots' control. They implemented a master-slave strategy, where the operator uses a local master haptic robot to control a remote mobile robot. A similar master-slave-based teleoperation method was implemented in [81] where researchers introduced a robot control concept for ergonomic bilateral robot control. [82] conducted a quantitative physical ergonomics assessment on two different robot control UI, including a standing interface with a whole-body motion capture system and a seated interface with a 3D mouse. In [83], authors applied

machine learning methods on a robot control framework that is capable of autonomously generating a user-adapted body-machine mapping function for drone operation.

In recent years, with the continuous advancement of VR technology, the performance and affordability of VR devices have significantly improved [84]. VR technology creates a virtual environment that allows users to engage in immersive interactions even in non-realistic settings. This characteristic aligns closely with the need for user-friendly human-robot interaction interfaces in robot control [85]. Consequently, there have been many research studies focused on integrating VR technologies with robotic control systems. Most of the consumer-grade VR devices contain their own input strategy for intuitive interaction in the virtual environment. Therefore, combining these input methods with robot control modules is a common practice. For example, [86] combined VR HMD with Touch X-based haptic interfaces to teleoperate a collaborative robot. The human interaction interface generates the desired pose of the robot end-effector in the virtual environment and transmits the control command to the real robot side. [87] proposed a task-centric VR robotic control system that focuses on the task itself rather than the direct manipulation of robotic hardware. [88] proposed a control method for Toyota Human Support Robot. The robot head motion and the robot arm motion are aligned with the motion of VR HMD and VR controller, respectively. The head motion control suffers from huge latency, though, which may cause uncomfortable situations such as feeling sick. In [89], researchers use a

haptic glove to control a multi-DoF robot arm while also obtaining haptic feedback.

2) Robotic Control Awareness

In robotic control systems, it is important to enable the robot to provide some significant information for the operator to perceive the remote space and make better decisions. The term situation awareness was first emerged from aviation psychology, but it is applicable broadly to any cognitive activity and information processing, including robot control [90]. In robotic control systems, awareness refers to the human operator's capability to perceive and understand the environment around the remote robot [91], which requires the operator to process a large amount of robot sensor data from the manipulation scene in real-time. However, many existing robot control systems do not provide significant feedback for human operators, resulting in less intuitive control and less comprehensive perception of the manipulation scenarios [73, 76, 92].

In recent years, researchers have proven that robot control awareness, such as reconstruction and visualization of robot's surrounding environment, haptic feedback, and force feedback is important for human-robot interaction in robot control systems [93]. Many research studies have been conducted to address the issue of awareness feedback provision in robot control systems. Among these research studies, the majority of them focus on scene visualization, which refers to displaying environmental context to the operator

through the interaction interface. Many 2D GUI-based interfaces rely on 2D image stream collected via RGB cameras (including those discussed in Section 2.2.1). For example, in [74], a GUI monitor is implemented to display the 2D view of the robot manipulation scene while also achieving object detection. Similarly, in [94], researchers performed object segmentation based on the image stream captured from an RGB camera deployed on the dual-armed robot. [80] proposed an orthographic vision-based teleoperation system by visualizing the remote environment and providing depth perception information via a single webcam. In [95], the first-person view and third-person view of the dual-armed robot are displayed on the monitor, as well as the robot pose state via visualization of the robot's real-time 3D model. In [82], researchers proposed a robot interface that provides raw image-based visual feedback from the remote robot for operators via 2D GUI interface.

Recently, many efforts have been made to explore and identify the best way to represent the physical world in virtual environments. [96] presented a telepresence approach to merge visual information from multiple cameras based on a VR interface. In this system, researchers utilized a static global stereo camera, and a local RGB-D camera mounted on the end-effector of the robot, and they only rendered the local scene with point cloud to provide the essential information for manipulation. By having both the scene and the geometric objects in the same space they determine which object is occluded and present the corresponding effect to the user in the virtual scene. [97] utilized a Kinect RGB-D camera to obtain the depth information of the

surrounding environment and rendered the whole scene in the virtual space through real-time mesh reconstruction in large-scale indoor and outdoor environments. In [98], researchers implemented a remote RGB-D camera as a wrist camera of the UR5 robot manipulator and provide video and depth feedback in real-time. This visualization feedback is rendered in the virtual world based on Unreal engine, including the point cloud information of the manipulation scene and raw image stream displayed in front of the user view. [99] adopted a similar strategy by rendering 3D point cloud information as well as 2D raw image from multiple views, including a first-person view, captured from a webcam installed on the robot gripper, and several third-person views. [100] established a virtual room where several virtual video displays are deployed at certain locations, where the VR user's brain infers the 3D representations directly, rather than having a GPU or CPU interpret the data to 3D and then back into images for each eye. In [88], the motion of the head of implemented humanoid robot is horizontally aligned with the VR HMD. A stereo camera is installed on the robot head, so when the human operator tries to move the head around to view the scene, the robot head will also move around to enable the 3D camera to capture the corresponding scene. Nevertheless, although the researchers attempted to utilize the capability and the functionality of the robot as much as possible to achieve natural human-like movement of the robot in robot control, there is still latency between human and robot movement. Considering the visualization method, it is possible to cause uncomfortable situations for the user. [101] proposed a category-agnostic scene-completion

algorithm that segments and completes individual objects from depth images. Although they did not implement any VR-related devices, their work can be easily integrated with VR since their virtual scene was developed based on Unity game engine. In [102], each manipulated object was applied with a tag. When one of the cameras in the workspace detects a tag, the ID is looked up, and the corresponding virtual representation of the object is displayed in the scene. [103] introduced a robot control interface that collects, processes, transfers, and reconstructs the immersive scene model of the workspace from point cloud in VR based on a deep neural networks (DNN) method.

Additionally, many efforts have been made to evaluate the effectiveness of different visualization strategies. For example, in [104], researchers evaluated how using virtual features, such as a 3D robot model, object target poses, or displaying distance to a target, affects operator performance in completing teleoperation pick-and-place tasks. [105] compared an immersive 3D visualization to a standard 2D video-based visualization and found that by displaying real-time 3D scene information, the ability to self-localize in the scene, avoid obstacles and control the visualization view is improved. In [106], researchers investigated the influence displaying different levels of environmental information has on task performance and operator situation awareness in VR robotic control interfaces. They found that the time to complete the task is reduced when displaying full information compared to the representative model, while accuracy remains the same between both.

Nevertheless, cognitive load demand is much higher during full information visualization.

2.3 Research Gaps

In previous sections, we have reviewed existing literature and provided an overview of the current advancements in LLM task planning and VR robotic control in the human-centric manufacturing scenarios. While notable progress has been made, several limitations have emerged during the review process. These gaps are summarized and discussed in this section to highlight the challenges that remain and to motivate further research.

2.3.1 LLMs for Robot Task Planning

Recent advancements in LLMs have demonstrated their significant potential in transforming robot task planning by bridging the gap between natural language instructions and robotic control. Unlike traditional rule-based and learning-based approaches, LLMs excel at interpreting high-level human instructions, reasoning over complex contexts, and generating structured task plans. Studies have explored their application in zero-shot task planning, hierarchical decomposition of goals, and integrating multimodal inputs, such as combining language and vision for enhanced contextual understanding. These efforts highlight the ability of LLMs to adapt to diverse scenarios and

simplify human-robot interaction, offering promising solutions for dynamic and unstructured environments. However, several challenges persist, limiting their practical application in real-world scenarios. Most existing approaches struggle to directly generate low-level robot-executable commands, often requiring additional manual processing or intermediate layers to refine abstract plans. Furthermore, many methods lack a structured and hierarchical framework to systematically handle the complexity of task planning, leading to fragmented outputs. Additionally, while LLMs have shown promise in general robotics, their application in specific industrial contexts, such as manufacturing and production assembly, remains underexplored. These domains demand domain-specific solutions capable of addressing intricate workflows and ensuring precision, adaptability, and scalability. Addressing these gaps is essential for advancing LLM-based robot task planning in practical applications.

To address these challenges, this thesis develops a multi-layer LLM-based robot task planning method that converts human natural language instructions into executable robot commands, which will be discussed in Chapter 3.

2.3.2 VR-based Intuitive Robotic Control

Research on robot control has made significant progress in recent years. Robot control systems aim to establish intuitive interfaces for human operators to

control robots while providing comprehensive environmental information. Traditional approaches often rely on 2D graphical interfaces and button-based input devices, such as keyboards, joysticks, or touchscreens. While these methods are user-friendly, they are not intuitive enough for robot control, especially in complex and dynamic scenarios. Recent efforts have explored more natural control methods, such as gesture recognition, motion capture, and haptic feedback, to enhance user experience. Additionally, the integration of VR technology has provided immersive environments and improved visualization through 3D reconstruction, depth information, and real-time point cloud rendering, enabling better spatial awareness and task performance. Nevertheless, traditional robot control approaches still lack intuitive interaction and feedback methods, while some VR-based systems, despite their advancements, are still not sufficiently intuitive for seamless human-robot interaction. These limitations hinder the operator's ability to effectively control the robot and perceive its environment, emphasizing the need for further improvements in control interfaces and feedback mechanisms.

To address these challenges, this thesis proposes a VR-based robotic control system that enables intuitive and seamless robot control and comprehensive visual awareness, which will be discussed in Chapter 4.

3 LLM-based Robot Task Planning

The integration of LLMs into robotic task planning has gained significant attention due to their ability to process natural language instructions and generate logical task plans [51, 52, 53, 54]. While existing research has shown promise in understanding high-level human language instructions and commands, several challenges persist. Notably, most current approaches struggle to effectively translate high-level human instructions into low-level, executable robot commands that align with the requirements of dynamic and complex manufacturing environments. This limits the practical application of LLMs in real-world robotic systems, where precision and adaptability are critical. In this chapter, an LLM-based robot task planning approach is presented, which enables efficient and effective translation from high-level tasks into low-level robot commands, ensuring seamless integration of human guidance and robot execution.

3.1 Introduction

In the era of smart manufacturing, robots have become indispensable for improving productivity, precision, and flexibility in dynamic production environments. As human-centric manufacturing becomes increasingly common, the ability for industrial robots to understand and execute tasks efficiently is

critical to achieving seamless integration into complex workflows. Central to this capability is robot task planning, which involves translating high-level instructions into robot actions that robots can perform in real-world scenarios. Normally, traditional task planning methods often rely on predefined rules or manual programming, which require significant time and expertise to implement. While these approaches are effective in static and predictable environments, they struggle to adapt to the complex and uncertain nature of highly unstructured manufacturing environments. Such inability has limited their scalability and flexibility, creating a pressing need for more advanced and adaptive task planning solutions.

Recent advancements in LLMs have unlocked new possibilities for improving robot task planning. LLMs are advanced systems trained on vast amounts of textual data to understand, generate, and interact with human language. LLMs excel at tasks involving natural language processing (NLP), including text generation, translation, summarization, and question answering. Models like ChatGPT have demonstrated remarkable performance across various domains.

With their exceptional natural language understanding and reasoning capabilities, LLMs can process high-level human instructions expressed in natural language and generate meaningful responses. This makes them particularly suitable for bridging the gap between human operators and robots, enabling intuitive communication and task delegation. In the context of task planning,

LLMs offer the potential to decompose complex instructions into structured steps and generate robot-executable commands. Early research studies have shown promise in applying LLMs to translate textual instructions into task plans, demonstrating their ability to handle diverse and unstructured inputs. These efforts aim to simplify human-robot interaction by reducing the reliance on manual programming and rigid rule-based systems [55, 56, 57, 58]. Nevertheless, current approaches often face significant limitations when it comes to the ability to seamlessly translate high-level instructions into precise, low-level robot commands. These methods typically produce generalized plans or abstract task descriptions, requiring additional layers of processing or manual intervention to convert them into executable actions.

To address this issue which significantly limits the practical application of LLMs in real-worlds robotic systems, in this chapter, we present a multi-layer LLM-based robot task planning method that converts human natural language instructions into executable robot commands. A multi-layer structure is implemented to improve the performance of robot commands generation. This method is presented aimed at bridging the gap between high-level tasks and low-level robot codes.

3.2 LLM-based Multi-Layer Robot Task Decomposition Planning

In this section, the proposed LLM-based multi-layer robot task decomposition planning framework is designed to bridge the gap between high-level natural language instructions and low-level executable robot control code. To better demonstrate the role of LLMs in understanding complex environments, we specifically select industrial component assembly/disassembly process as the research scenario. To achieve this, the framework employs a three-layer hierarchical structure that incrementally processes inputs and generates outputs, transforming human intent into precise robotic actions.

The framework begins with the Task Decomposition layer, where a high-level natural language instruction, supplemented by an image input, is processed. This layer allows the LLM to understand the overall task context by combining linguistic information with visual cues. The result is a series of decomposed sub-steps in natural language, each detailing a specific action required to complete the task.

Next, the Motion Description layer takes the decomposed sub-steps and generates precise positional relationships between objects involved in each specific task. This layer ensures that the task is not only broken down but

also contextualized in terms of spatial relationships, which are critical for robotic manipulation.

Finally, the Robot Command Coding layer translates these positional descriptions into executable robot control code. By leveraging robot primitive knowledge and external libraries, this layer converts abstract spatial relationships into concrete Python scripts that robots can execute.

The overview of the proposed solution is demonstrated in Fig. 3.1. The framework provides a seamless pipeline that integrates multi-modal inputs and systematically transforms them into robotic actions. In this work, we implement a multi-modal pre-trained LLM GPT-4o as the interface to convert human instructions into robot actions. This approach is aimed at enhancing the adaptability, efficiency, and precision of robot task planning in complex manufacturing scenarios. The following sections will provide a detailed discussion of each layer in the proposed framework.

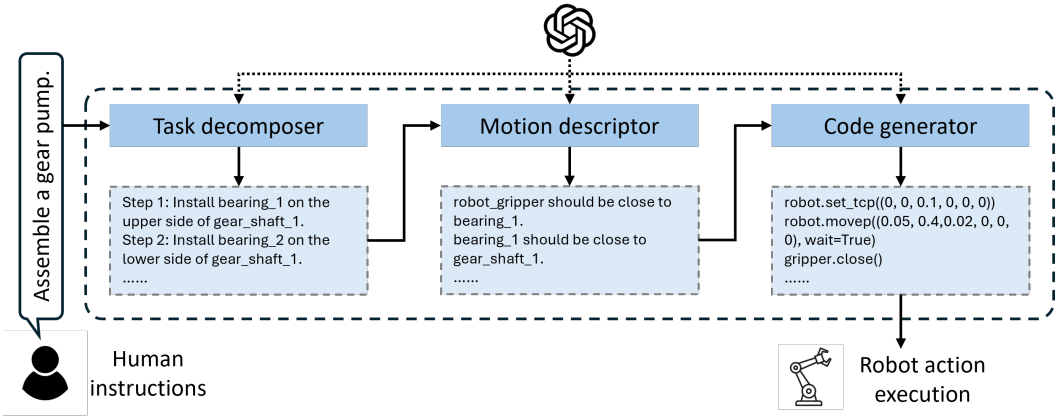

Fig. 3.1: LLM-based and human-prompt guided task planning and robot code generation.

3.2.1 Task Decomposition

The task decomposition layer is the first step of the proposed framework, aimed at breaking down a high-level task into a sequential list of specific sub-tasks. In this layer, the LLM is guided through carefully designed prompts to ensure it generates actionable and logically ordered task steps. To achieve this, both textual and visual inputs are utilized to provide comprehensive contextual information about the task, the environment, and the objects involved. The output is a sequence of sub-tasks, where each step specifies a discrete action involving two objects. The reason for involving only two objects is to ensure that each step includes a specific target point coordinate when generating executable code.

The prompt provided to the LLM is designed to include key details about the task environment and objects. This includes information such as object types, identifiers, their spatial coordinates, and their attachment relationships, such as “The gear shaft with the bearings installed should be placed inside the pump body, with gear_shaft_1 positioned on the left side and gear_shaft_2 on the right side inside the pump body”. Additionally, visual input enhances the LLM’s ability to contextualize the task by offering a clear depiction of the components and their spatial arrangement, as shown in Fig. 3.2. The prompt also contains sample tasks and corresponding answers to help the LLM learn the desired decomposition logic.

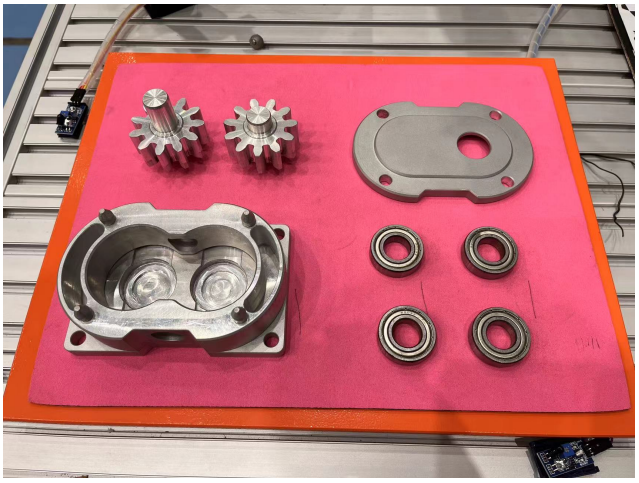

Fig. 3.2: Gear pump components used in the assembly task.

To ensure clarity and consistency, specific rules are embedded in the prompt to limit the output format. Specifically, each output line is restricted to describing actions involving a maximum of two objects. This restriction ensures that the subsequent layer, which focuses on motion description, can effectively process the output and describe spatial relationships for each step independently.

3.2.2 Motion Description

The motion description layer is designed to translate each sub-task output from the task decomposition layer into detailed descriptions of the target spatial relationships between objects within the task environment. This layer is designed aimed at bridging the gap between task-level instructions and robot-level operations by reformulating human-intuitive instructions into precise positional descriptions that robots can interpret and act upon. These

spatial descriptions form the basis for generating executable robot commands in the next layer.

The input to this layer consists of the decomposed sub-steps generated by the task decomposition layer. Each sub-step typically involves the robot gripper and two objects, focusing on their spatial relationships and interactions. The output is a natural language description of these relationships, specifying precise object alignments, placements, or relative positions.

To ensure consistency and precision, carefully designed prompts are employed to guide the LLM in generating the expected outputs. The prompts include task-related information, such as the objects' identifiers, their properties, and the overall task context. A template is provided to standardize the formulation of spatial relationships, along with several examples to demonstrate the expected output format. This approach ensures the LLM can infer spatial relationships from the task context while maintaining logical and structured outputs. Additionally, specific rules are integrated into the prompt to constrain the layer's output.

3.2.3 Robot Code Generation

This section provides the final stage of the proposed framework, tasked with converting the spatial relationship descriptions generated in the motion description layer into low-level robot executable code. This step is critical

for bridging the gap between abstract task planning and concrete robotic actions by leveraging given information including predefined robot primitive skills, third-party libraries, and a structured prompt design. The primary goal of this layer is to translate high-level instructions into precise, executable commands that align with the robot's capabilities and hardware.

The input to this layer consists of natural language descriptions of spatial relationships between manipulated components, while the corresponding output is Python code that directly interacts with the robot's control system. To achieve accurate and actionable code generation, the prompt for this layer incorporates several key components. First, it provides a comprehensive list of robot's primitive skills, which define the available control APIs of the robot, such as robot motion and gripper action control. Second, the prompt specifies the third-party Python libraries (e.g., NumPy) that can be used to extend functionality. These APIs and libraries serve as the building blocks for generating executable robot code. Third, the prompt includes a structured output template to ensure consistency and readability. This template organizes the code into clear, step-by-step instructions with appropriate function calls, making the code both executable and interpretable. Then, the prompt provides several example inputs and outputs, demonstrating and helping guide the LLM toward producing outputs that adhere to the expected format and logic. Finally, to ensure reliability and functionality of the generated code, specific rules are provided in the prompt. These rules constrain the output to include

only necessary commands, avoid redundancies, and maintain compatibility with the robot's control system.

Table 3.1: Robot primitive skills and corresponding APIs.

Primitive Robotic Skills	**Python APIs**
move to a coordinate	rob_movep((coord), wait)
open the gripper	gripper_open()
close the gripper	gripper_close()

Table 3.1 provides an overview of the primitive APIs utilized in this layer, which are implemented through the Python interface provided by the robot's driver. These APIs support the execution of fundamental robotic actions and serve as the foundation for constructing complex behaviors. They focus on motion and manipulation commands, while abstracting away the trajectory planning and hardware-specific details. One of the key advantages of using these APIs is the modularity they offer, which means more complex actions such as pick-and-place can be constructed. This allows the LLM to focus on generating logical action sequences rather than dealing with the intricacies of motion control or hardware constraints. Furthermore, while the current set of APIs are designed for basic pick-and-place tasks, additional functions can be integrated to extend the robot's capabilities, enabling the LLM to generate action plans for more complex assembly/disassembly scenarios. An example of generated codes is illustrated in Fig. 3.3, where the instruction from previous layer is to pick the gear shaft 2 and place it into the pump body.

```python
import numpy as np

def main():
    # Coordinates (assumed based on the image and task):
    # - gear_shaft_2 is at (0.1, 0.2, 0.1, 0, 0, 0)
    # - pump_body is at (0.3, 0.2, 0.1, 0, 0, 0)
    # - Above gear_shaft_2: z = 0.2
    # - Above pump_body: z = 0.2

    # Move gripper above gear_shaft_2
    rob_movep((0.1, 0.2, 0.2, 0, 0, 0), wait=True)
    # Descend to gear_shaft_2
    rob_movep((0.1, 0.2, 0.1, 0, 0, 0), wait=True)
    # Close gripper to pick up gear_shaft_2
    gripper_close()
    # Lift gear_shaft_2
    rob_movep((0.1, 0.2, 0.2, 0, 0, 0), wait=True)

    # Move above pump_body
    rob_movep((0.3, 0.2, 0.2, 0, 0, 0), wait=True)
    # Descend to pump_body
    rob_movep((0.3, 0.2, 0.1, 0, 0, 0), wait=True)
    # Open gripper to place gear_shaft_2 inside pump_body
    gripper_open()
    # Lift the gripper away from pump_body
    rob_movep((0.3, 0.2, 0.2, 0, 0, 0), wait=True)

if __name__ == "__main__":
    main()
```

Fig. 3.3: Example executable code generated by code generator.

3.3 Experimental Results

To quantitively evaluate the effectiveness of the proposed three-layer robot task planning approach, in this section, we conduct a series of experiments designed to compare its performance against alternative methods. We predefined a manipulation task, “assemble a gear pump”, as the original input for the task planner. To evaluate the performance of our proposed method, two baseline methods are chosen comparably, including: i) a single-layer robot code generator (that directly translate the original task into robot code) without image input; ii) a three-layer robot code generator without image input. Additionally, we also adopted three different pretrained LLM agents, including GPT-4o, GPT-4, and GPT-3.5 for comparative study. Each configura-

tion is evaluated using the same predefined assembly task, and the generated robot action codes will be evaluated by human experts, and the criterion for determining the success of an experiment is whether the plan correctly calls functions in a logical and feasible sequence. For each configuration, the evaluation metric is the success rate, which is calculated as the percentage of 20 trials where the generated results met the criteria of human expert evaluation.

Table 3.2: Results of LLM-based task planning.

Method	GPT-4o	GPT-4	GPT-3.5
Single-layer prompt (no image)	4/20	6/20	0/20
Three-layer prompt (no image)	12/20	9/20	2/20
Three-layer (image input) (ours)	20/20	-	-

The experiment results are shown in Table 3.2. From the results, we can see that the proposed three-layer robot task planner with multi-modal input achieved higher success rates than baseline methods. The hierarchical multi-layer design outperformed the single-layer structure, which struggles to decompose tasks and generate coherent robot code directly from high-level instructions. In terms of LLMs, GPT-4o has higher success rates among different methods. This may be due to its larger scale of parameters. Additionally, the implementation of multi-modal input also improves the performance of the planner notably.

The experimental evaluation of the LLM-based robot task planner provides an initial demonstration of how LLMs can be applied to complex robotic assembly tasks, showcasing their potential to enhance task planning and

execution in human-centric manufacturing scenarios. While the current system remains in an early stage and faces limitations such as dependency on computational resources and challenges in real-time adaptability, the results highlight the promising role of LLMs in bridging natural language instructions and robotic control. With further optimization and integration into real-world robotic platforms, LLMs are likely to enable future manufacturing systems with greater intelligence and adaptability, paving the way for more seamless and autonomous manufacturing workflows.

3.4 Chapter Summary

The integration of LLMs into robotic task planning has introduced a novel framework for enhancing human-centric manufacturing by leveraging natural language understanding and reasoning capabilities. This chapter proposes a three-layer LLM-based planning approach for generating executable robot code in complex assembly tasks, with a focus on structured task decomposition. The main contributions of this chapter can be summarized as follows: 1) Developed a hierarchical three-layer planning framework leveraging task decomposition, motion description, and robot code generation to bridge high-level human instructions and low-level robotic execution. 2) Designed a meticulously crafted prompt structure tailored for each layer, incorporating task-specific domain knowledge, robotic primitive skills, and multi-modal input integration, enabling the planner to generate logical and executable

robot actions. Experimental evaluations demonstrate favorable performance of the proposed approach compared to alternative methods, highlighting the importance of hierarchical reasoning and multi-modal inputs. The results validate the system's ability to generate accurate and logical robot code for complex tasks, showcasing its potential for real-world human-centric manufacturing scenarios.

Despite its performance, some limitations remain, such as the computational cost of multi-layered reasoning and response latency associated with LLMs. To address these challenges, future research may explore the development of more efficient, task-specific approaches tailored to industrial applications. Additionally, enhancing the integration of other data modalities could further improve the system's robustness and adaptability to diverse manipulation scenarios.

4 Intuitive Robot Control for Complex Environments

In complex and dynamic manufacturing environments, ensuring precise and efficient robotic manipulation remains a significant challenge, particularly in human-centric manufacturing systems. Traditional robotic control methods often rely on pre-programmed instructions or rigid interfaces, which lack adaptability and demand extensive expertise from human operators. These limitations hinder the flexibility required for robots to handle diverse and unstructured tasks in real-world scenarios. In recent years, advancements in VR technology have opened new possibilities for intuitive robot control by providing immersive and interactive interfaces. However, existing VR-based teleoperation systems often face challenges such as high operator cognitive load and inefficient integration between human input and robot execution.

To address these issues, in this chapter, we present a VR-based robot teleoperation interface to enable intuitive and immersive robotic control for complex manufacturing environments. By leveraging immersive VR interfaces and real-time data exchanging mechanisms, the proposed system enables seamless interaction between human operators and robotic manipulators, facilitating flexible and efficient task execution in human-centric manufacturing scenarios.

4.1 Introduction

Robotic control systems are supposed to facilitate human-robot interaction even when the operator is a non-professional user [69]. Interaction interfaces that are not enough intuitive can impact the cognitive load of human operators, resulting in significantly increased learning costs. In such cases, the efficiency of performing teleoperation tasks with robots can be compromised. This can lead to decreased performance, longer task completion times, and potentially higher error rates. Intuitive interfaces, on the other hand, are designed to align with the operator's mental model and expectations, making it easier for them to understand and control the robot. By reducing cognitive load, the interfaces enable operators to focus more on the task at hand, leading to improved efficiency and effectiveness in robotic teleoperation scenarios.

The cognitive load primarily arises from the complexity and intuitiveness of the interaction interface [107]. Traditional robot monitoring and teleoperation systems typically employ 2D graphical interfaces to display the current status and relevant information of the robot. They rely on input devices such as keyboards, mice, or touch-based devices to capture instructions from operators. While 2D graphical interfaces can effectively present textual or graphical information, they often fall short when it comes to conveying depth information about the environment or work scene. Furthermore, traditional input devices struggle to establish a clear mapping between the operator's in-

tent and the robot's actions, especially when dealing with complex high-DoF robots or intricate end-effector structures.

Facing these limitations, there is a need for an interactive approach that can provide a more intuitive representation of the robot's working environment. This requires an interaction interface that goes beyond the capabilities of 2D graphical interfaces and can more accurately depict the dynamic changes in a 3D space. However, most existing research in this area has been primarily developed based on 2D graphical interfaces. Obtaining dynamic spatial information requires operators to perform redundant operations. Furthermore, there is a need for a more direct method of inputting commands. This interaction approach should better reflect the operator's direct intent, rather than requiring the translation of control intentions into operations within a graphical interface before being transmitted to the physical robot. However, most existing robot teleoperation systems still rely on using simple input devices to accomplish potentially complex tasks, which can increase the operator's cognitive load in terms of learning and adaptation.

Nowadays, with the continuous advancement of VR technology, the performance and affordability of VR devices have significantly improved [84]. VR technology creates a virtual environment that allows users to engage in immersive interactions even in non-realistic settings. This characteristic aligns closely with the need for user-friendly interaction interfaces [85]. In this context, the research interest in VR-integrated human-robot teleoperation

has been increasingly growing [108]. Compared to traditional 2D graphical interfaces, VR environments offer users a more intuitive perception of spatial depth, allowing for a more comprehensive environmental awareness. VR enables users to explore and interact within virtual spaces as if they were in physical environments. This reduces their cognitive load, as they no longer need to spend significant time extracting crucial spatial depth information from flat graphics. Furthermore, input devices with spatial tracking capabilities in VR allow users to quickly locate target objects or positions, which is challenging to achieve in 2D graphical interfaces. This enhances the efficiency and accuracy of user interactions within the virtual environment.

Another important issue to be considered is the system's compatibility with different robot platforms. Robots from different manufacturers, such as UR, KUKA, Franka Emika, often have different communication protocols and interfaces. This makes it challenging to port and adapt interaction modules designed for one certain type of robot to another. This issue becomes particularly pronounced with multi-DoF collaborative robots, where replacing a robot would require significant modifications to the system's configuration, resulting in substantial workloads. Therefore, a fundamental communication framework that is applicable to multiple robot platforms is necessary.

In view of the above, the aim of this research is to propose a VR-based teleoperation interface to enable intuitive and immersive human-robot teleoperation. The proposed system mainly contains two major parts, including:

1) a VR interface, which is developed based on Unity engine; and 2) a robot control interface, which is developed based on Robot Operating System (ROS) [109]. ROS is an open-source robotics middleware with great flexibility and feasibility, which enables a wide range of applications for robot development [110]. Therefore, our proposed robot control framework is easy to port to another commercial robot which is supported by ROS. In addition, we implement more intuitive interaction methods than existing teleoperation systems which provide better experience for human operators.

4.2 VR-based Robot Teleoperation System

This section provides a comprehensive explanation of the proposed VR-based robot teleoperation system, which is designed to enable intuitive and precise control of robotic manipulators in complex environments. The system collects two primary inputs: target robot poses commands provided by the human operator, and real-time visual feedback provided by the RGB-D camera. These inputs allow the operator to interact seamlessly with the robot by issuing direct control commands while maintaining situational awareness of the task environment. At the remote side, the generated control commands are executed by a robotic manipulator equipped with an end-effector, ensuring accurate manipulation of objects in real time, as shown in Fig. 4.1.

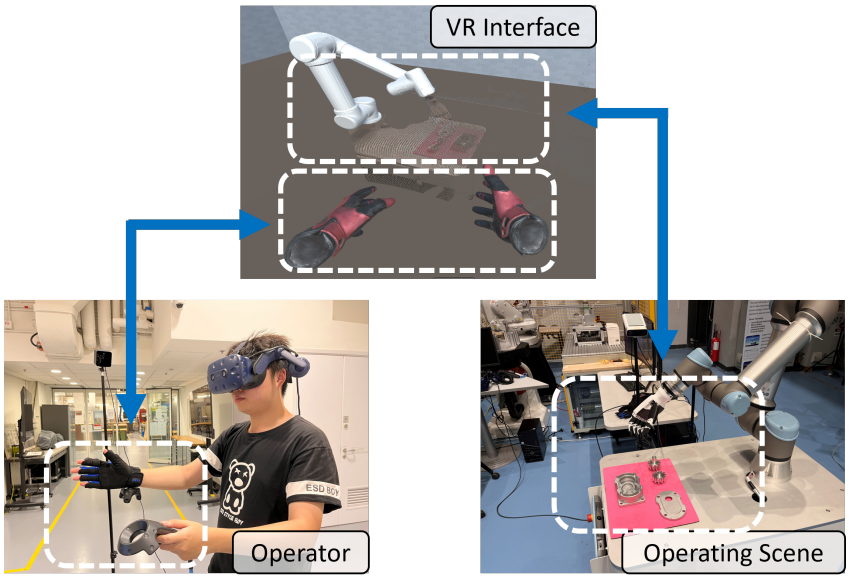

Fig. 4.1: An operator teleoperating a robot via the proposed system. The work area information is captured and visualized in the virtual environment, where a virtual representation of the robot arm reflects the real robot pose.

The system is developed using the Unity 3D engine and ROS, which together provide a robust framework for integrating software modules and hardware components. External hardware devices are basically categorized into input and output devices. Through this seamless integration of software and hardware, the proposed system achieves a high degree of flexibility and precision, bridging the gap between human intention and robotic execution in complex manufacturing tasks. In the following sections, the components of the proposed VR teleoperation system will be introduced individually.

4.2.1 Overall System Design and Implementation

The overall framework of the proposed VR teleoperation system is illustrated in Fig. 4.2. The entire system consists of four major components, including input signal collection, VR-based interaction module, motion planner module

and execution and scene monitoring. In the input signal collection phase, the operation commands are collected through a set of input devices, basically including a glove-based controller and a joystick controller. The human operator holds the input devices and provides control commands. The glove-based controller captures the operator's finger joint pose information, while the joystick enables the operator to trigger robot target pose commands by pressing buttons on the controller. These command inputs, as well as the work area's image stream captured by the on-site RGB-D camera, are transmitted to the VR interaction module.

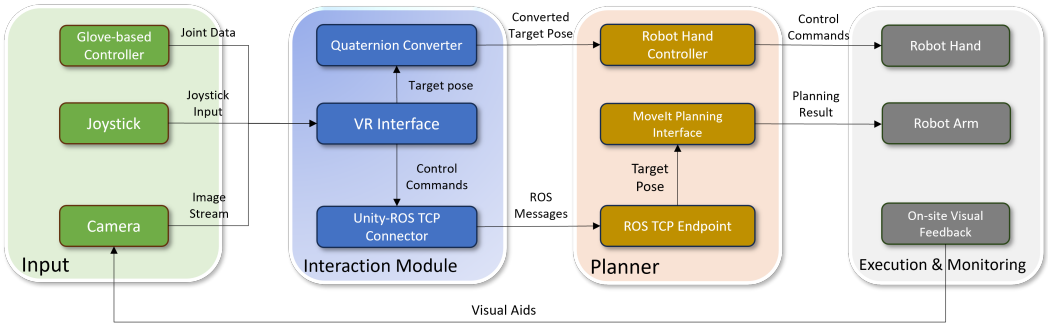

Fig. 4.2: Framework of the proposed VR teleoperation system.

The interaction module is designed to process and convert the input data into expected format. The image stream data is processed and rendered on a plane in the virtual environment to visualize the remote scene in the VR interface. The finger pose information is converted to Euler angles using the quaternion converter, while the robot target pose information is serialized and transformed into ROS message format which can be processed by ROS. These target poses are regarded as the robot hand target pose and robot arm target pose, respectively, and are then forwarded to the motion planner

module, where direct control commands for the robot hand and robot arm are generated for execution.

4.2.2 Interaction Module

The interaction module is responsible for receiving and processing the data collected via the input devices to enable human-machine interaction in a virtual environment. There have been numerous research studies focused on the interaction mode of teleoperation systems [69]. In our proposed system, the interaction mode is designed based on VR interface.

The VR interface of the proposed teleoperation system is developed using the Unity 3D engine, a versatile and widely adopted platform for creating interactive 3D environments [111]. Unity provides a comprehensive suite of tools and features designed to support the development of VR and AR applications across various platforms, including mobile devices, desktops, and web-based systems. Its robust scripting API enables developers to write custom code in C#, allowing for precise control over the behavior of virtual objects and the implementation of complex interaction mechanisms. Furthermore, Unity offers an extensive library of assets, scripts, and plugins, which significantly streamline the development process and enhance system functionality. The engine's real-time rendering capabilities allow for realistic simulations of object interactions, a critical feature for creating an effective and intuitive human-machine interaction interface. These capabilities make Unity an ideal

platform for the development of the VR teleoperation system, ensuring both flexibility and realism in the virtual environment.

1) Input Devices Integration

A virtual interaction environment was developed within the Unity platform, integrating VR components and real-world environment visualization. The VR hardware utilized in this project consists of a consumer-grade VR kit, including a VR headset and a pair of joysticks. Additionally, a glove-based controller was incorporated to further enhance the teleoperation experience by providing more intuitive and precise interaction capabilities. These devices were seamlessly integrated into the Unity scene using the OpenVR toolkit [112], which facilitates compatibility and communication between the hardware and the virtual environment. Within the virtual scene, both the joystick and the glove-based controller are accurately represented and visualized, offering a realistic and immersive teleoperation interface. This setup enables operators to interact intuitively with the virtual environment, bridging the gap between human intention and robotic execution.

The input devices are modeled as prefabs within the Unity environment and rendered in the 3D virtual scene to ensure accurate representation. The position and orientation of the joystick controller and glove controller are tracked using laser sensors from the base station, with their poses integrated into the Unity environment via the OpenVR plugin. Hand gestures made by the operator are captured through IMUs embedded in the glove controller,

enabling the virtual hand model to mirror the physical movements of the operator's hand. This functionality provides an intuitive and immersive experience, allowing the operator to interact naturally with virtual objects through hand gestures.

In addition to the glove controller, the joystick controller features a touchpad and several buttons, each mapped to specific functions. These functions include controlling the robot's end-effector movement in the horizontal plane or adjusting its vertical position. When the operator presses a button, a new goal position message is generated and sent to the motion planning module implemented within the ROS framework. This communication is facilitated through a TCP connection, enabling efficient and reliable data exchange between the Unity environment and the robotic control system. This setup ensures seamless integration of operator inputs and robot motion, enhancing the intuitiveness and precision of teleoperation.

In terms of work environment visualization in the virtual interface, the proposed system leverages an RGB-D camera to stream real-time RGB and depth information from the workspace, enabling point cloud visualization in the Unity virtual environment. In this work, the system processes the RGB and depth data into a point cloud representation. These streams are subscribed to as topics from the camera and passed as textures to the Unity virtual interface. To achieve this, for each corresponding pixel, a colored point is generated in the virtual scene. The point's color is derived from the associated RGB pixel,

while its position is calculated based on the depth value from the RGB-D sensor, thus allowing real-time visualization of the remote workspace. To integrate the RGB-D camera and the robot base coordinate, it is necessary to establish a precise alignment between the two coordinate systems. The alignment process involves determining the rigid transformation, including a rotation matrix R and a translation vector T.

In the left-handed coordinate system used by virtual robot in Unity VR, the positive Z-axis points forward, whereas in the right-handed coordinate system used by ROS, the positive Z-axis points outward from the screen in the opposite direction. To reconcile this difference, the camera coordinates are first transformed into a pseudo-right-handed coordinate system by inverting the Z-axis of each point:

$$p'_c = [x_c, y_c, -z_c]^T, \tag{4.1}$$

where p_c represents a point in the original camera coordinate system and p'_c is the adjusted point in the pseudo-right-handed coordinate system.

After addressing the coordinate system discrepancy, the rigid transformation between the camera and robot coordinate systems is determined. Let $P_c = p_c1, p_c2, \ldots, p_cn$ denote a set of points in the camera coordinate system and $P_r = p_r1, p_r2, \ldots, p_rn$ denote the corresponding points in the robot's

coordinate system. The relationship between the two systems is modeled as:

$$p_r = R \cdot p'_c + T, \tag{4.2}$$

where $R \in \mathbb{R}^{3\times3}$ is the rotation matrix and $T \in \mathbb{R}^3$ is the translation vector. Using the points collected above, the rigid transformation is estimated by minimizing the alignment error between the two coordinate systems. The rotation matrix R is derived from the covariance of the point sets through singular value decomposition, while the translation vector T is determined by comparing the centroids of the two sets.

The final transformation from the camera coordinate system to the robot coordinate system is given as:

$$p_r = R \cdot [x_c, y_c, -z_c]^T + T. \tag{4.3}$$

To validate the calibration, the mean alignment error is calculated:

$$\text{Error} = \frac{1}{n} \sum_{i=1}^{n} \left\| p_{ri} - (R \cdot p'_{ci} + T) \right\|. \tag{4.4}$$

If the error exceeds a predefined threshold, recalibration is performed using additional corresponding points or improved measurement precision. The point cloud effects in the virtual environment before and after calibration are shown in Fig. 4.3.

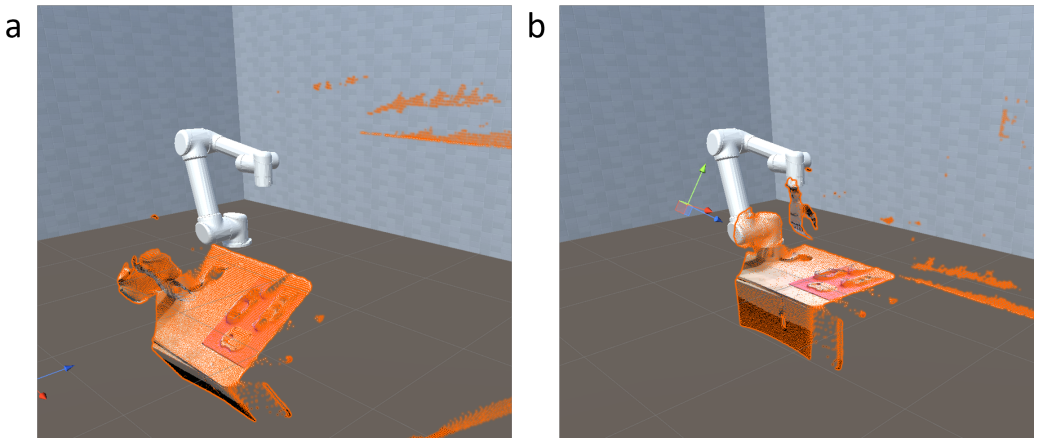

Fig. 4.3: Point cloud data collected by RGB-D camera before and after calibration: (a) before calibration; (b) after calibration.

2) Unity-ROS Integration

To ensure seamless communication and integration between the virtual environment and the ROS framework, the proposed teleoperation interface incorporates a robust communication mechanism between the Unity engine, implemented as a .Net application, and ROS. Given that ROS operates as a node-based communication framework reliant on data exchange between Python- or C++-based nodes, the communication protocol is designed to bridge the gap between ROS nodes and the C#-based Unity scripts. This integration is achieved through the utilization of an open-source software library, Unity Robotics Hub [113]. By leveraging the capabilities of this tool, the system establishes reliable TCP connections to facilitate real-time data

exchange between Unity, where the VR interface is developed, and ROS, which serves as the foundational platform for robot control. This architecture enables efficient and synchronized communication, ensuring that commands and feedback are transmitted seamlessly between the virtual interface and the robotic system.

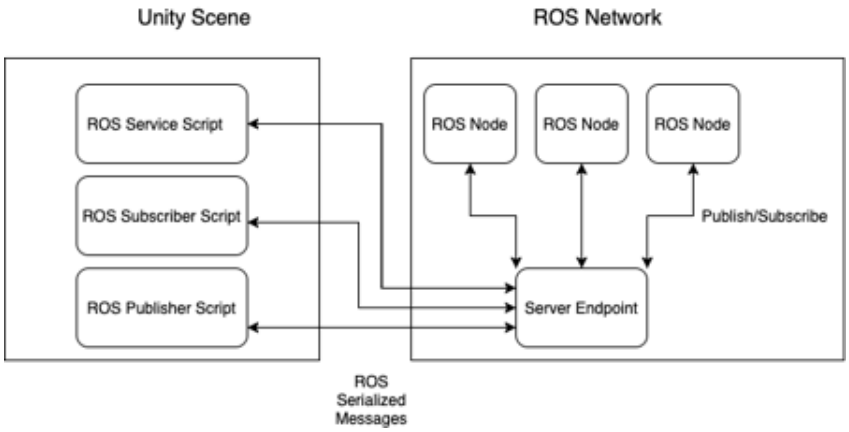

Fig. 4.4: Overview of the framework of our Unity-ROS communication established based on TCP connections [113].

An overview of the Unity-ROS communication framework, established using TCP connections, is presented in Fig. 4.4. Once the connection is successfully established, a TCP server endpoint operates as a ROS node, enabling seamless bidirectional message exchange between Unity and ROS. For example, when the human operator initiates a new goal position command for motion execution, the target position is transmitted from a publisher script on the Unity side to the ROS server endpoint. The message is then forwarded to the designated ROS node, joint_state_controller, within the robot motion control module, where a motion trajectory is generated and executed by the robot arm. Similarly, when the robot reaches a new position, its joint state information is collected and published by the ROS node, joint_state_publisher,

to the server endpoint. This data is subsequently transmitted to the Unity environment, where it is processed to update the virtual robot model to reflect the robot's new pose. This mechanism ensures real-time virtual-physical mapping, enabling real-time synchronization between the physical robot and its virtual counterpart in the teleoperation system.

To establish the communication framework, a ROS connection script component is integrated and assigned to an empty object within the Unity scene. Concurrently, the aforementioned TCP server endpoint is configured to enable bidirectional message exchange between Unity and ROS, allowing efficient and effective communication between the virtual environment and the robotic system.

One important aspect to consider is the serialization of messages within the Unity framework since the original messages generated by C# scripts in .Net applications are not able to be processed directly by ROS services. To address this, we utilize a message generation plugin that generates C# classes from the ROS messages and serialize messages being passed to ROS framework as ROS would internally serialize them. This generation plugin is responsible for serialization and deserialization functions from ROS-based messages, which are transmitted via .msg files, ensuring seamless communication between the two systems.

In addition, a specific set of scripts serves as the Unity counterpart of ROS nodes which function as service, subscriber and publisher. Each script is

attached to an empty object within the Unity scene, thus the message publishing or subscribing function runs once a frame when the project is working. These components utilize the ROS connection script and TCP endpoint node to establish communication channels with the corresponding ROS nodes. Publishers are responsible for transmitting data generated within the virtual environment to the ROS system, while subscribers receive and process data from ROS nodes. Service providers facilitate the exchange of more complex requests and responses between Unity and ROS, enabling advanced functionalities within the collaborative manufacturing process.

3) Robot Arm Transform Conversion

To enable the robot arm to follow the operator's hand movements, a transform conversion mechanism is implemented to translate human input into the robot's end-effector target positions. This process begins with the capture of the operator's wrist position using the VR input devices and laser tracking system within the Unity environment. The VR system provides real-time spatial pose data, including the position of the operator's wrist, which serves as the primary input for controlling the robot arm's motion. The wrist position data is first analyzed to calculate its relative positional changes over time. These changes, representing the operator's hand movements, are used to generate motion commands for the robot.

In Unity, the coordinate system is left-handed, while in ROS, the coordinate system is right-handed. To transform coordinates between Unity and ROS,

the axes need to be remapped due to their differences in handedness and axis orientation. The transformed positional data is then passed to the ROS framework as a target pose for the robot's end-effector. Specifically, this target position is sent to the motion planning module, where a collision-free trajectory is computed, ensuring safe and precise motion of the robot arm. By continuously updating the target pose based on the operator's hand movements in the VR environment, the robot arm is able to follow the operator's motion in real time.

4) IMU-based Human Hand Motion Detection

The input method utilized by the human operator to control the robot plays a critical role in determining the intuitiveness and efficiency of the operation. For end-effectors with complex articulated structures and a high DoF, traditional control methods—such as buttons or touch interfaces—often prove cumbersome and unintuitive, leading to a significant increase in the operator's cognitive load. In this work, the robot's end-effector is a five-fingered robotic hand, which requires precise and natural control to perform complex tasks. To address this challenge, we employ a glove-based controller equipped with joint-mounted IMU sensors, enabling the operator to control the robotic hand through intuitive hand gestures. This approach provides a more natural and user-friendly interface for high-DoF robotic manipulation.

The glove-based controller is integrated with 11 joint IMU sensors which represent 15 joints of human hand and capture the joint poses in quater-

nion format, three on each finger and one on the wrist. The joints that can be detected are defined as PINKY DISTAL, PINKY INTERMEDIATE, PINKY PROXIMAL, RING DISTAL, RING INTERMEDIATE, RING PROXIMAL, MIDDLE DISTAL, MIDDLE INTERMEDIATE, MIDDLE PROXIMAL, INDEX DISTAL, INDEX INTERMEDIATE, INDEX PROXIMAL, THUMB DISTAL, THUMB INTERMEDIATE, THUMB PROXIMAL, respectively, as shown in Fig. 4.5.

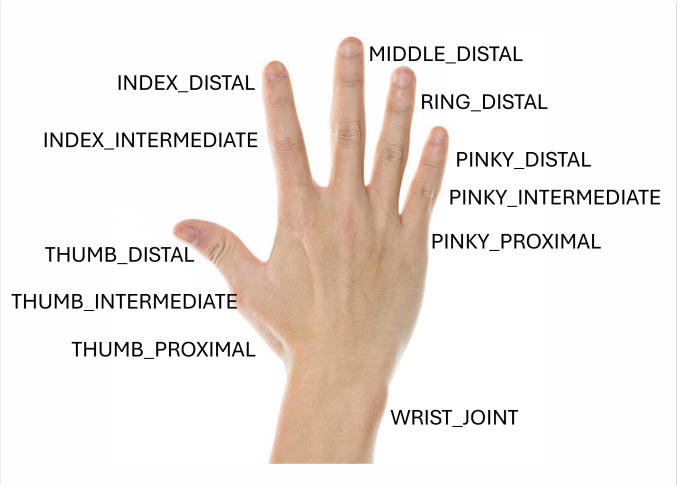

Fig. 4.5: Joint definitions of the human hand.

The VR interface receives joint pose information from the glove-based controller in the form of quaternions provided by the IMU sensors. However, this raw quaternion data does not directly convey the human operator's hand gestures. To address this, a hand pose processing module was developed to extract meaningful gesture information. This module processes joint pose data and calculates the degree of flexion for each finger, thereby determining the operator's hand gesture. Specifically, a quaternion converter is employed to compute the angle of separation between two spatial pose quaternions representing adjacent joints.

Given the limited DoF of the robotic hand, the system focuses only on the distal joint of each finger and the wrist joint. Each finger, from the pinky to the thumb, is assigned a unique index (finger 1 to finger 5, respectively). To simplify the control process and align with the DoF constraints of the robotic hand, the system prioritizes the distal joint of each finger as the primary input parameter for gesture recognition. For instance, if the pose quaternion of the wrist and the pose quaternion of the distal joint of the finger i $(i \in 1, 2, 3, 4, 5)$ are denoted as $\mathbf{q}_0$ and $\mathbf{q}_i$ respectively:

$$\mathbf{q}_0 = (qw_0, qx_0, qy_0, qz_0), \quad \mathbf{q}_i = (qw_i, qx_i, qy_i, qz_i). \tag{4.5}$$

The angle θ_i between the two IMUs is calculated as:

$$\theta_i = 2 \times \cos^{-1}\left(|\mathbf{q}_0 \cdot \mathbf{q}_i|\right), \tag{4.6}$$

and θ_i will be processed as the flexion degree of finger i and forwarded to the robot hand controller. When the human operator flexes their fingers, the glove-based controller returns hand joint pose information and calculates the degree of flexion for each finger. This information is then processed and executed in the robot hand controller, causing the robotic hand's fingers to bend to the corresponding angles. This synchronization enables the operator's

hand gestures to be replicated by the robotic hand, achieving a coordinated motion between the operator's hand and the robotic hand.

4.2.3 Motion Planning Module

To enable the operation of the robot arm from the virtual environment, we rely on ROS as the fundamental framework for the robot's motion planning module. ROS is a widely used open-source framework that offers an extensive range of software libraries and tools specifically designed for the development and control of robotic systems.

By leveraging ROS, we can take advantage of its node-based architecture, which plays a crucial role in facilitating seamless communication between different programs. This architecture enables efficient and reliable data exchange over the ROS TCP network, ensuring smooth interoperability between the teleoperation interface and the physical robotic manipulator. Additionally, ROS provides robust and user-friendly APIs that support the development of custom nodes in widely used programming languages such as C++ and Python. This flexibility allows for the integration of diverse software modules, enabling ROS to serve as a versatile intermediary framework.

ROS is widely utilized across various mechanical domains, including industrial robotics, collaborative robots, and beyond, due to its exceptional versatility and adaptability. Rather than being tailored to a specific robotic

system, ROS functions as an integrated software platform that incorporates a broad range of libraries. These libraries provide a foundational framework for developing and managing diverse robot-related programs and hardware devices. By offering a standardized low-level communication protocol and a flexible logical architecture, ROS facilitates seamless integration and interoperability between different components of robotic systems. The core node-based communication framework of ROS, which underpins its modular and scalable design, is illustrated in Fig. 4.6. This architecture enables developers to construct robust and adaptable robotic applications.

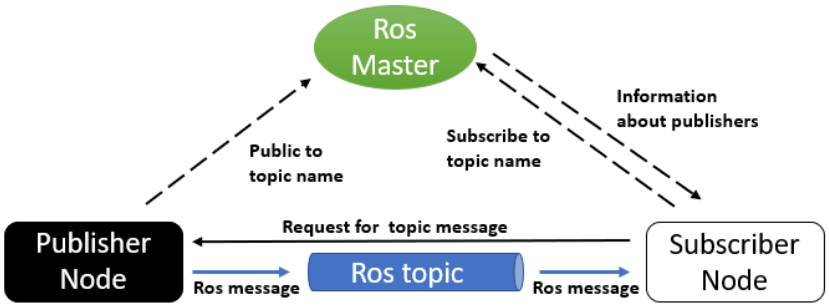

Fig. 4.6: Fundamental node-based framework of ROS communication. [114]

1) Robot Arm Control

The motion planning interface utilized in this study is developed based on MoveIt [115]. MoveIt stands as the primary choice among motion planning packages that function within the ROS framework [116]. It offers a comprehensive set of tools and generic interfaces that cater to a wide range of robots, empowering them to perform motion planning tasks while ensuring collision avoidance capabilities.

As MoveIt is integrated in ROS, it follows the node-based foundational communication framework. MoveIt provides high-level system architecture for receiving motion planning requests and generating motion waypoint execution solutions. MoveIt works based on the URDF format of the robot model and provides collision detection functionality through its integrated path planning algorithms. In the case where the physical robot matches the model, MoveIt can generate path planning results.

A notable feature of MoveIt is its Move Group Interface, which serves as the primary execution node of the high-level system architecture and provides a user-friendly functionality for executing robot operations, as shown in FigMoveGroup. This interface allows researchers to easily access the robot controller and the motion planning scene, streamlining their interaction with the robotic system. The move_group node uses the ROS param server to get three kinds of information, including URDF, SRDF and MoveIt configuration, and talks to the robot through ROS topics and actions. Also, it communicates with the robot to get current joint state information and to talk to the controllers on the robot.

MoveIt works with motion planners through a plugin interface. This allows MoveIt to communicate with and use different motion planner from multiple libraries. The interface to the motion planners is through a ROS action or service. In our proposed planning framework, the Open Motion Planning Library (OMPL) is seamlessly integrated within the MoveIt interface [117].

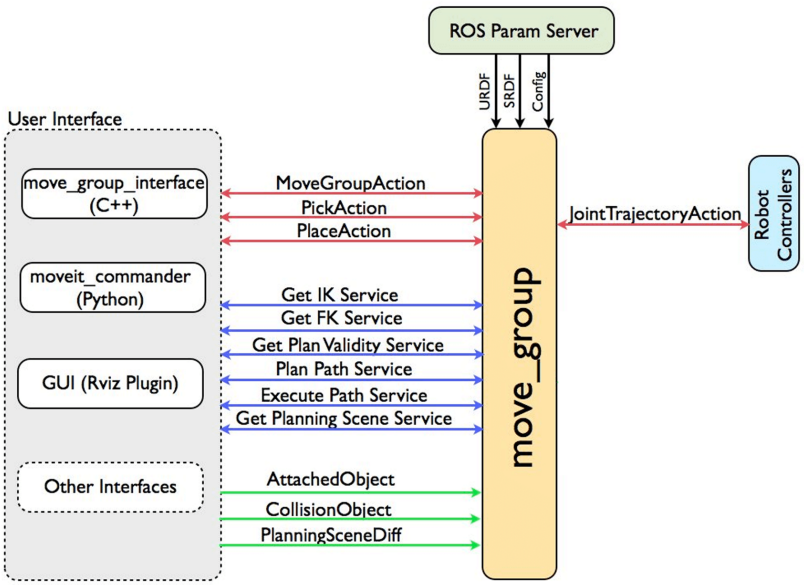

Fig. 4.7: Overview of MoveIt's Move Group interface. [115]

This integration enables OMPL to generate motion planning solutions, which can then be executed by MoveIt to guide the robot towards the desired goal position. The MoveIt environment acts as the backend, providing the necessary support for abstract planners within OMPL to address various robot motion planning problems effectively. The library is designed so it can be easily integrated into systems that provide the additional needed components, such as ROS.

Additionally, OMPL offers a wide array of state-of-the-art sampling-based planning algorithms to cater to different scenarios and requirements. In this study, we employ the Rapidly-exploring Random Tress Connect (RRT-Connect) algorithm as the robot motion planner [118]. RRT-Connect is an extension of the concept of Rapidly-exploring Random Trees (RRT), which is a random sampling-based algorithm that employs a tree structure for exploring the configuration space. While traditional RRT algorithms can

suffer from slow convergence, RRT-Connect addresses this issue through the introduction of a greedy expansion approach. As shown in Fig. 4.8, the depicted random tree $T1$ is situated on the left side, while the random tree $T2$ is positioned on the right side. Here, x_init denotes the starting point of the path planning procedure, x_rand represents a randomly sampled point, x_near signifies the node in the search tree that exhibits the closest distance to x_rand, and x_new indicates a newly expanded node along a fixed search step length $steplen$, where x_near serves as the parent node. A search expansion encompasses two significant steps: random expansion and greedy expansion. The random tree $T1$ performs the random expansion, adhering to the conventional RRT algorithm. Conversely, the random tree $T2$ engages in the greedy expansion. For the greedy expansion, a heuristic function is employed as the greedy function. Specifically, $T2$ generates a novel node, x_new', in the direction defined by the angle between the target point x_goal and the newly generated node x_new from $T1$, while maintaining the same fixed step length $steplen$ as $T1$. Subsequently, $T2$ adopts x_new' as the parent node, progressively creating additional nodes towards the node x_new of $T1$, with the information of each new node being overwritten into x_new'. This process continues until an obstacle is encountered or until the successful connection of the two random trees is achieved. If the greedy expansion fails to establish a connection between the two random trees, subsequent search expansions necessitate the modification of their respective expansion approaches. These modifications introduce potential alterations to their strategies for future search expansions.

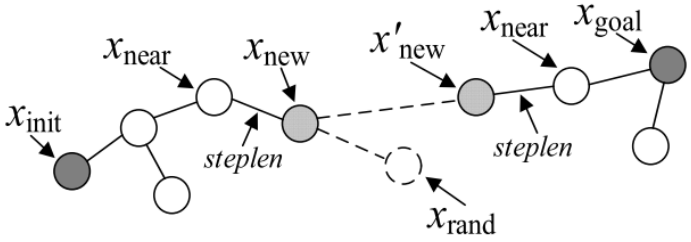

Fig. 4.8: RRT-Connect planning algorithm [119].

By leveraging the capabilities of MoveIt and integrating OMPL's planning algorithms, our framework enables effective motion planning for the robot arm. The Move Group Interface within MoveIt provides an intuitive interface for researchers to specify high-level goals and constraints, while OMPL's planning algorithms, particularly the RRT-Connect, efficiently explore the configuration space to generate feasible motion plans.

The overall framework of the proposed motion planning interface is depicted in Fig. 4.9. The process begins with MoveIt acquiring the robot's current joint pose information, which is used to compute both the current pose of the robot and the position of the end-effector, serving as the starting pose. When the operator issues a new motion command, the corresponding target pose is passed into the MoveIt environment. The starting and target poses are then processed within the motion planning interface, where a collision-free trajectory is computed using the RRT-Connect algorithm. If a feasible solution cannot be found within the specified time limit, an error is returned, prompting either a replanning attempt or a request for a new target pose from the operator. Once a valid motion trajectory is successfully generated, it is transmitted to the robot joint controller via ROS action communication.

The controller then executes the motion plan, driving the physical robot to the specified position. Upon reaching the target pose, the motion command is considered completed, and the system awaits the operator's next command.

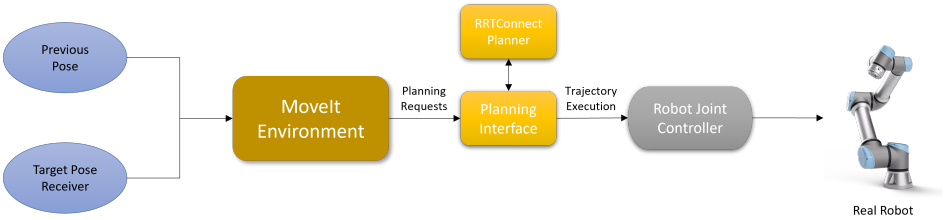

Fig. 4.9: Proposed robot motion planning module in the teleoperation system.

As depicted in Fig. 4.2, our robot arm motion planning interface receives motion requests (operator commands) from the ROS endpoint, which is connected to the VR interaction interface. However, it does not mean that our robot can only perform replicating the human operator's instructions. Due to the features of ROS and ROS-based software like MoveIt and other motion planning methods, we can easily publish external command messages that are aligned with the format predefined in our system. In other words, this interface serves as a middleware for connection between the operator and the physical robot manipulator while the command source can be easily changed. Therefore, future research can be easily built upon this robot manipulator motion planning module to explore more advanced topics, such as multi-modal human-robot interaction and robot learning from demonstration.

2) Robot Hand Control

To achieve more intuitive robot control, a 6-DoF five-finger robotic hand was implemented and attached to the robot manipulator as the end-effector. The

robotic hand is equipped with six linear servo drivers, which utilize RS485 communication interfaces to enable precise and reliable control. These servo drivers provide high torque output, allowing the robotic hand to generate sufficient force for performing a variety of grasping tasks. In addition, the robotic hand is integrated with built-in pressure sensors that enable it to detect resistance applied to the fingers by measuring the current passing through the motors. This functionality allows the hand to measure and respond to the force exerted during grasping. By configuring adjustable thresholds, the grip strength can be tailored to match the hardness or fragility of the objects being manipulated. This feature significantly enhances the robot hand's versatility and adaptability, enabling it to handle a wide range of objects with varying physical properties.

The main control unit interacts with the robot hand by reading from and writing to its internal registers for status retrieval and control. Reading from registers refers to the upper-level system retrieving the values of the internal registers in the robot hand. The upper-level system sends a read command to the robot hand, including the starting address and length of the register group to be read where a group refers to a set of adjacent registers. Upon receiving and successfully verifying the data, the robot hand sends back the corresponding register data to the upper-level system. Similarly, writing to registers involves the upper-level system writing corresponding data to the internal registers of the robot hand which can be done in groups. The upper-level system sends a write command to the robot hand, containing the starting

Table 4.1: Command frame format for writing to the robot hand's registers.

	Value	Description
byte[0]	0xEB	Header
byte[1]	0x90	Header
byte[2]	Hands_ID	Robot hand ID
byte[3]	Register_Length+3	Length of the data frame
byte[4]	0x12	Write register command flag
byte[5]	Address_L	Lower byte of starting address
byte[6]	Address_H	Higher byte of starting address
byte[7] ... byte[7+Register_Length-1]	Data[0] ... Data[Register_Length-1]	Data to be written to the registers
byte[7+Register_Length]	cheksum	Checksum

address of the register group and the data to be written. After receiving and successfully verifying the data, the robot hand sends a confirmation signal back to the upper-level system. The command frame format for writing to the robot hand's registers is shown in Table 4.1.

To facilitate data transmission from the operator to the robot hand, a Python-based upper-level interface is implemented. We take the angle value angle of each degree of freedom as the input to the function, which corresponds to the bending degree of the fingers. Through a series of transformations, the input angles of each degree of freedom are adjusted to satisfy the input range of the function. The input angles of each degree of freedom are then packaged and processed to convert them into the required bytes, as we talked about earlier in this section. These bytes are subsequently transmitted to the lower-level

controller through the established serial communication. Similarly, we can also retrieve current motor status, current magnitude for force sensing, and other information from the robot hand's lower-level controller using a similar approach through our upper-level program.

Due to the inherent limitations of IMUs used in glove-based controllers, such as their accuracy shortcomings and susceptibility to magnetic field interference, the original angle θ_i often exhibits noticeable instability. This instability can adversely affect the teleoperation of the robot hand, leading to imprecise and jerky movements. To address this challenge, a dedicated robot hand controller is implemented to process the angle θ_i obtained from the quaternion converter. This controller generates direct control commands for the robot hand, thereby ensuring smoother and more precise teleoperation.

To achieve stable and reliable control, a moving average filtering algorithm is introduced into the controller [120]. This algorithm effectively mitigates the impact of instability on the control signals. By applying the moving average filter, fluctuations and noise in the angle measurements are smoothed out, resulting in more reliable and consistent control commands. However, it is important to consider the specific characteristics of each finger's joint motion when applying the filter. The thumb, for instance, typically has a relatively smaller range of motion compared to the other fingers. To ensure that the thumb of the robot hand does not move too slowly during operation, the filter window size is adjusted specifically for the thumb. By reducing the

window size, the filter becomes more responsive to changes in the thumb's angle, allowing for more dynamic and agile movements. On the other hand, the filter window size for the other fingers remains larger, as their range of motion is typically greater. Mathematically, if we denote the window size of the filter for finger i as n_i, the actual output angle value at time t for finger i, denoted as $\theta_(i_t)$, can be calculated using the moving average filtering algorithm as follows:

$$\theta_{i_t} = \frac{\theta_{i_{t-1}} + \theta_{i_{t-2}} + \cdots + \theta_{i_{t-n}}}{n_i}. \tag{4.7}$$

The resulting filtered angle values are then transmitted to the upper computer interface of the robot hand. At this interface, the filtered angles are further processed and translated into target position parameters for each finger. This ensures that the motion of the robot hand corresponds accurately to the motion of the operator's hand, enabling intuitive and synchronized control. Consequently, the operator's hand movements can correspondingly control the motions of the robot hand, as shown in Fig. 4.10, achieving more smooth and intuitive remote control.

Similar to the previously discussed robot arm, our robot hand also features an open upper computer interface. This implies that, in addition to direct control by human operators, we have the flexibility to input operational commands through alternative means to accomplish the control of the robot

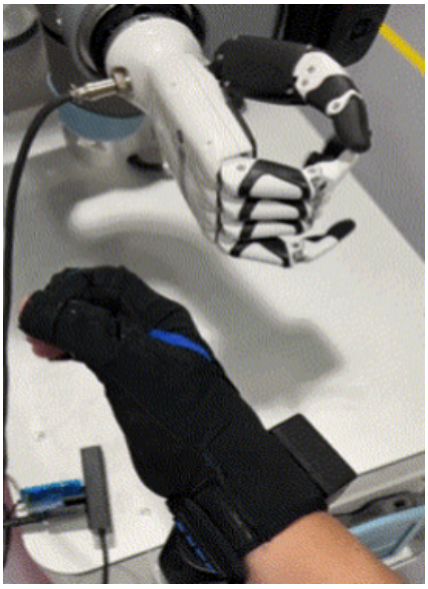

Fig. 4.10: Human and robot hand motion mapping.

hand's movements. Consequently, our robot hand control interface exhibits scalability, allowing for future research in areas such as robot learning from demonstration to be conducted based on our teleoperation system.

Thus far, we have discussed the control methods for both the robot arm and the robot hand. Our system enables the operation devices to be intuitively operated by human operators while also providing expandability for further research and the addition of modules, among other possibilities.

4.3 User Study

This section is aimed at further validating the effectiveness and feasibility of our proposed VR-based robot teleoperation system. To achieve this, we conducted an empirical user study to test our system. We assumed that our VR teleoperation system could facilitate intuitive robot control and could

enhance operation efficiency and experience. Therefore, several participants were invited to use the system and conduct certain tasks, after which they were asked to fill in a questionnaire to express their feelings about this approach.

4.3.1 Experimental Setup

To conduct the experiment, the UR5e collaborative robot was utilized, chosen for its versatility and reliability [121]. The UR5e features a 6-DoF robotic manipulator, a teach pendant, and a control box, making it well-suited for a wide range of robotic applications. To enable object manipulation, a 6-DoF five-finger robotic hand was mounted as the end-effector of the manipulator. The experimental system operated on a Linux PC running the ROS environment, which was connected to the UR5e's control box via an Ethernet connection. Additionally, the robotic hand was interfaced directly with the Linux PC through the RS485 communication protocol.

For the VR component, the HTC VIVE Pro system was employed, comprising a VR head mounted display (HMD) and a tracked joystick controller [122]. In addition, a five-finger glove-based controller equipped with multiple IMU sensors was utilized, with a VR laser tracker installed on the wrist part to capture its spatial position. These VR devices were modeled and rendered within the Unity 3D environment, running on a Windows PC. To enhance scene visualization, an RGB-D camera, Kinect Azure [123], was deployed

to capture the assembly workspace from the side angle, providing real-time point cloud feeds that were integrated into the virtual environment.

To compare the proposed VR-based teleoperation system with traditional robotic control methods, a baseline approach, robot teach pendant, was selected for the experiment. Teach pendant is a standard interface for collaborative robots that allows operators to program and control the robot through a handheld device with buttons and a touch screen. Since this method does not utilize an intuitive input mechanism similar to the glove-based controller proposed in our system, the robot's end-effector was replaced with a conventional two-finger gripper during this approach.

In terms of participants, 9 participants were invited to act as human expert operators to complete a gear pump assembly task using both two control methods. We first introduced our method to each participant, including the use of VR HMD and input devices, robot control mechanisms, and the system workflow. Then, each participant took part in the gear pump assembly task and completed the task 3 times per method. After completing the task, they filled in a 5-point Likert scale evaluation questionnaire with 8 questions.

4.3.2 Results and Analysis

Some representative operation processes are presented in Fig. 4.11, and the questions and questionnaire results are shown in Fig. 4.12. Overall, the

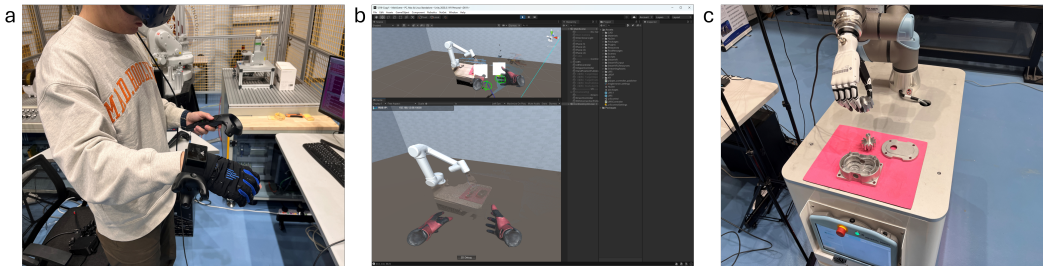

Fig. 4.11: Representative sample results of the user study. (a) A participant is operating using VR equipment. (b) The Unity virtual environment interface during the operation process, with the bottom section showing the operator's perspective in VR. (c) Grasping components during the gear pump assembly operation.

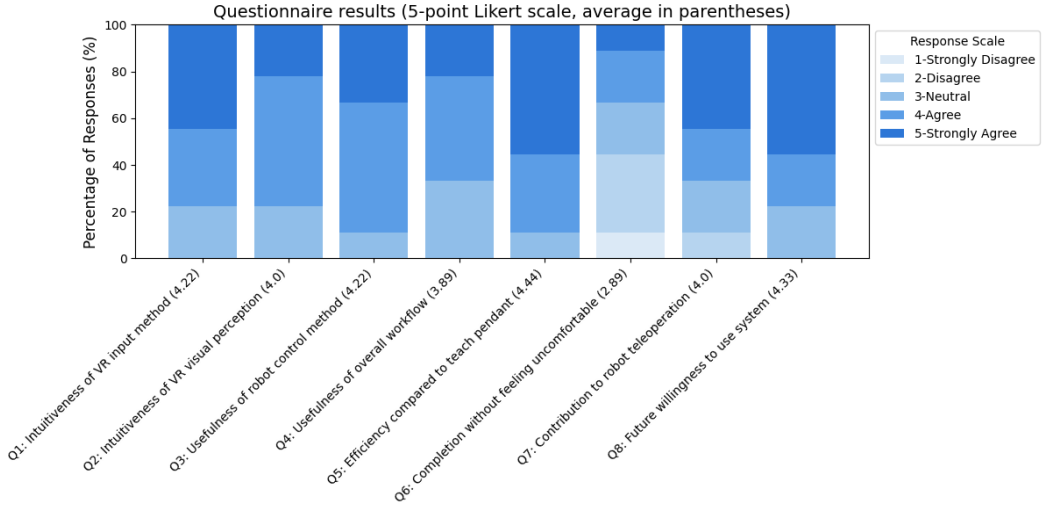

Fig. 4.12: Questionaire results (5-point Likert scale, average in parentheses).

results of the questionnaire reflect the general effectiveness of the proposed method, with participants expressing predominantly positive attitudes toward the system. Specifically, participants generally found the VR input scheme to be intuitive, which may be attributed to the strategy of mapping the robotic arm's end-effector movements to hand motions, and the use of a more intuitive glove-based input method with the mapping to a five-finger robotic hand. Additionally, the use of a VR headset, combined with depth cameras for point cloud data collection, provided visual feedback that was relatively easy to comprehend. The overall experience of robot control and task completion

using the system was reported as satisfactory, with efficiency significantly exceeding that of traditional teach pendant-based control schemes, which rely on 2D screens and physical buttons. Most participants agreed that VR noticeably enhanced the user experience of robot teleoperation and expressed interest in adopting further optimized versions of the system in the future.

However, the feedback from participants also highlighted several issues. First, while the depth camera provided immersive 3D visual feedback, the relatively low density of the point cloud caused difficulties when the operator's virtual camera in the Unity environment moved too close to the point cloud, making fine details difficult to discern. This issue could be mitigated by adjusting the size of the point cloud. In addition, since only a single depth camera was placed on the side of the robot, operators could achieve relatively complete visual perception of the point cloud only from certain specific angles. In the future, multiple depth cameras may be placed around the robot and calibrated to achieve more comprehensive point cloud rendering and visual feedback. Furthermore, the robotic hand used in this experiment had limited degrees of freedom, making it challenging to execute precise grasping motions aligned with the operator's intentions and hand movements, thus constraining efficiency. Employing a more dexterous and higher-precision robotic hand could address this limitation, though it would come at the cost of increased expense.

The most significant challenge, however, was the discomfort caused by wearing VR equipment. Many participants reported experiencing dizziness when frequently changing view directions while operating in the virtual environment. Additionally, prolonged use of relatively heavy HMDs led to fatigue, exacerbating dizziness and discomfort. Future research could explore the use of more lightweight, all-in-one VR devices to alleviate these issues and design more user-friendly virtual environments to reduce the frequency of motion sickness occurrences.

4.3.3 Discussion

In the previous sections, we demonstrated that our proposed VR teleoperation approach can perform manufacturing assembly tasks effectively and efficiently. The interaction between human operators and robot manipulators is achieved via the VR interaction interface, robot motion planning interface, and the TCP connection between them.

Industrial manufacturing involves diverse equipment and robots. Therefore, the adaptability and scalability of a teleoperation system determines its ability to seamlessly transition from one manipulation environment to another, accommodating different tasks under varying equipment conditions. The VR interaction module of our proposed system is developed based on Unity, a versatile platform that makes it adaptable to various VR devices. Moreover, Unity possesses powerful rendering capabilities, enabling future

enhancements such as reconstructing the environment through the capture of 3D point clouds using depth cameras and rendering them within the Unity environment.

Besides, our robot motion planning module also demonstrates adaptability and extensibility, primarily due to its foundation on the open-source framework ROS. ROS provides standardized interfaces and communication protocols, making it compatible with numerous collaborative robots, industrial robots, and other devices that offer corresponding ROS interfaces. The node-based communication mechanism in ROS facilitates the easy addition of new modules to the system. Additionally, our system utilizes the MoveIt motion planning module, which is integrated with ROS. MoveIt offers open interfaces and the flexibility to customize the choice of path planning algorithms, even allowing for the use of proprietary or custom algorithms. This flexibility ensures the scalability of the robot motion planning module to accommodate different robot types and diverse path planning requirements.

Despite the promising prospects offered by the proposed teleoperation system, there are still some limitations. One notable issue is the potential for VR-induced motion sickness, which can affect some users due to the mismatch between visual and vestibular cues in the immersive environment. Additionally, prolonged use of the VR headset can lead to physical discomfort and fatigue, particularly due to the weight of the device and the strain it places on the operator's neck and eyes. These factors may impact the

operator's performance and reduce the overall usability of the system over extended periods, highlighting the need for further ergonomic and hardware improvements.

Furthermore, although the open-source nature of ROS and ROS-based software packages offers convenience for development and enhances system adaptability, it can also introduce challenges related to instability, inefficiency, and versioning issues. For instance, MoveIt provides control and joint information reading interfaces that are applicable to most robots, making it an integral component of the motion planning module in our proposed teleoperation system. However, the instability of MoveIt path planning often leads to planning failures in practical operations, resulting in the inability to execute action commands provided by operators. Additionally, different robot manufacturers may support different versions of ROS-based software, which can pose challenges during the integration of different devices.

4.4 Chapter Summary

In this research, we propose an intuitive VR-based robotic control approach for human-centric manufacturing tasks that enables human operators to teleoperate robots in real-time. A key feature of the approach is the development of a teleoperation interface that seamlessly integrates with the ROS framework, facilitating efficient communication and data exchange between the virtual environment and ROS-enabled robotic manipulators. The sys-

tem integrates input devices designed to provide operators with an intuitive control experience, combined with immersive visual feedback. This allows operators to naturally perceive the workspace environment and effectively control the robot, achieving higher efficiency in manipulation tasks.

Building on the work presented in this chapter, several potential research directions can be explored in the future to further enhance the proposed system. First, efforts can be made to optimize the operator's experience and comfort, such as by incorporating ergonomic improvements and adopting more advanced VR hardware with higher resolution and reduced weight. These enhancements could mitigate user fatigue and improve long-term usability. Second, refining the control algorithms to improve the precision and efficiency of robot operations, particularly for tasks requiring fine manipulation, offers another promising avenue for development. Additionally, the system can serve as a foundation for research into learning from demonstration (LfD), where data collected during teleoperation could be utilized to train robots to autonomously perform similar tasks. These advancements would not only improve the overall performance of the system but also expand its applicability to a wider range of complex industrial tasks.

5

Conclusions

As manufacturing evolves towards a human-centric paradigm, the integration of human adaptability with robotic capabilities has become essential to meet the growing demand for flexible and personalized production. This paradigm emphasizes seamless interaction and harmonious coexistence between humans and robots, thus recognizing the critical role of perception, planning, and execution as interconnected processes in enabling robots to operate effectively in shared environments. In this thesis, through an extensive review of existing literature, key challenges have been identified in both robotic task planning and control methods, which often fall short of meeting the flexibility and adaptability required in dynamic manufacturing scenarios. Current task planning approaches struggle to bridge the gap between high-level human instructions and low-level robot action execution, while traditional robot control and teleoperation systems are limited by unintuitive interfaces and high operator cognitive loads. To address these issues, this thesis proposes two solutions, respectively: an LLM-based robot task planning approach that enhances the translation of human instructions into executable robot commands, addressing challenges in planning, and a VR-based intuitive robot control method that improves human-robot interaction in robotic control tasks, addressing challenges in perception and execution. In this chapter,

we summarize the key contributions and discuss the limitations and future directions of this research in the following sections.

5.1 Contributions

Contribution 1: A multi-layer LLM-based robotic task planning approach has been presented to bridge the gap between high-level human natural language instructions and low-level executable robot commands.

To address the challenges in the planning process of human-centric manufacturing scenarios, we explore robot task planning guided by humans in the context of human-robot interaction. Building on the limitations identified in existing research and drawing inspiration from advancing pre-trained LLMs, we proposed a three-layer LLM-based robot task planning framework. This approach takes natural language instructions as input, supplemented by visual assistance, and generates executable robot control code as the final output. We provided a detailed explanation of the framework's structure, including the specific prompt design methods employed in each layer. An experiment has been conducted to validate the feasibility and reliability of the proposed method.

Contribution 2: A VR-based robotic control system has been proposed to explore intuitive and seamless robot teleoperation and comprehensive visual awareness.

To tackle the issues of the perception and execution process in human-centric manufacturing, we focus on exploring a solution for enhancing user experience of robot control systems. Traditional robot control methods often struggle with unintuitive interfaces and limited visual feedback, leading to high operator cognitive loads. Therefore, in this work, VR technology has been introduced to develop an intuitive and immersive robot control interface. We design a VR-based teleoperation framework that combines virtual environments and real robot workspace. This integration enables operators to naturally control the robot by using intuitive input method, while also receiving comprehensive visual feedback in a virtual environment. Our framework also incorporates an efficient robot motion planning method to enable seamless robotic control. Experimental validation demonstrates the system's potential to enhance task performance and operator experience.

5.2 Limitations

Despite the contributions made in this study, several limitations remain to be addressed. Regarding the proposed LLM-based robot task planning method, our reliance on pre-trained LLMs introduces challenges such as dependency on a stable network connection and computational latency, which may hinder deployment in certain environments. Training models locally could mitigate these issues but would require additional computational resources. Although we implemented a vision-assisted multimodal input approach, the

current input method still has shortcomings, such as the lack of more comprehensive prior knowledge and environmental information. As a result, generating highly accurate robot codes often requires substantial human guidance. Therefore, the proposed method primarily focuses on task planning and bridging the gap between high-level and low-level commands, without fully addressing the challenge of obtaining precise environmental data. In the future, we plan to integrate computer vision-based modules to provide more accurate environmental information, such as precise object coordinates, which could enable the generation of more reliable and executable robot commands.

For the proposed VR-based robot control system, as discussed in Chapter 4, the existing limitations primarily stem from hardware constraints. Prolonged use of VR devices may cause discomfort, which could potentially be mitigated by adopting lighter and more ergonomic equipment. Although depth cameras were employed to provide point cloud information, the current visual feedback remains insufficiently detailed to offer precise operational feedback for fine-grained tasks. In the future, a multi-camera setup may be adopted to enhance visual perception; however, this approach could increase the complexity of on-site deployment and impose additional constraints. Furthermore, the experiments in this study were conducted over a local wired network, ensuring low-latency operation. However, the system has yet to be tested in internet-based or more complex wireless network environments.

Future work will include testing in such scenarios to evaluate the system's real-time performance and stability for remote control applications.

5.3 Future Research Directions

Human-centric manufacturing emphasizes harmonious and efficient collaboration and coexistence between humans and robots within shared workspaces. This study proposed an LLM-based robot task planning method and a VR-based robot control system, offering feasible solutions at the levels of perception, planning, and execution. However, the current approach still has several areas that require refinement and further development. In the final section of this thesis, we outline potential future research directions building upon the work presented in previous chapters.

(1) Integration of VR-Based Robot Teleoperation and Monitoring with LLM-Enhanced Task Planning and Execution Framework.

The next research direction of this study is to integrate the LLM-based robot task planning method discussed in Chapter 3 with the VR-based robot control system presented in Chapter 4. This integration aims to enable operators to leverage the assistance of pre-trained LLMs within a VR-based remote immersive workspace. Operators would have the flexibility to guide or control robots directly through intuitive VR inputs, facilitating natural and seamless robot operations. Alternatively, they could utilize the LLM-based

task planning method by providing natural language instructions to guide robots in autonomous task execution. Furthermore, real-time monitoring could be achieved through visual feedback within the virtual environment.

To enhance the VR interactive interface, additional elements could be introduced, such as incorporating quantified information from the operational site into the VR interface. This information could be presented through floating UI components to provide operators with feedback, enabling a more comprehensive perception and understanding of the overall scene. Achieving these functionalities would require a more robust and integrated multimodal system, making this integration the primary focus of our future research and development efforts.

(2) To propose a robot learning from demonstration system based on our VR teleoperation framework.

The VR-based robot interaction interface proposed in this study currently supports only direct control and perception. In future research, we plan to collect spatial motion data from operators using input devices for demonstration purposes, enabling robots to complete tasks through guided teaching. Additionally, machine learning methods, such as DRL, could be employed to allow robots to learn and imitate human behavior patterns, facilitating the evolution from passive control to autonomous task execution.

Moreover, the multi-layer framework adopted in our LLM-based task planning approach offers potential to support the learning-from-demonstration process at each layer. For instance, the task decomposition layer could provide task guidance to human operators, reducing the time and effort required during the demonstration process. Similarly, the code generation layer could offer initial execution plans for the robot, accelerating the training process of imitation learning.

Building on the research directions described above, along with other potential avenues of exploration, our overarching goal is to integrate the strengths of humans, robots, and AI-based methods to develop a more comprehensive human-centric manufacturing solution for the evolving landscape of smart manufacturing.

References

[1] Kendra Briken, Jed Moore, Dora Scholarios, Emily Rose, and Andrew Sherlock. „Industry 5 and the human in Human-Centric manufacturing“. In: *Sensors* 23.14 (2023), p. 6416 (cit. on pp. 1, 3).

[2] Baicun Wang, Fei Tao, Xudong Fang, et al. „Smart manufacturing and intelligent manufacturing: A comparative review“. In: *Engineering* 7.6 (2021), pp. 738–757 (cit. on p. 1).

[3] Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. „Large language models for robotics: A survey“. In: *arXiv preprint arXiv:2311.07226* (2023) (cit. on p. 1).

[4] Praveen Kumar Reddy Maddikunta, Quoc-Viet Pham, B Prabadevi, et al. „Industry 5.0: A survey on enabling technologies and potential applications“. In: *Journal of industrial information integration* 26 (2022), p. 100257 (cit. on p. 2).

[5] Sotirios Panagou, W Patrick Neumann, and Fabio Fruggiero. „A scoping review of human robot interaction research towards Industry 5.0 human-centric workplaces“. In: *International Journal of Production Research* 62.3 (2024), pp. 974–990 (cit. on pp. 2, 19).

[6] Andrea Teresa Espinoza Perez, Daniel Alejandro Rossit, Fernando Tohme, and Oscar C Vasquez. „Mass customized/personalized manufacturing in Industry 4.0 and blockchain: Research challenges, main problems, and the design of an information architecture“. In: *Information Fusion* 79 (2022), pp. 44–57 (cit. on p. 3).

[7] Yuze Jiang, Zhouzhou Huang, Bin Yang, and Wenyu Yang. „A review of robotic assembly strategies for the full operation procedure: planning, execution and evaluation“. In: *Robotics and Computer-Integrated Manufacturing* 78 (2022), p. 102366 (cit. on p. 3).

[8] Riccardo Gervasi, Federico Barravecchia, Luca Mastrogiacomo, and Fiorenzo Franceschini. „Applications of affective computing in human-robot interaction: State-of-art and challenges for manufacturing“. In: *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 237.6-7 (2023), pp. 815–832 (cit. on p. 3).

[9] Yeseung Kim, Dohyun Kim, Jieun Choi, et al. „A survey on integration of large language models with intelligent robots“. In: *Intelligent Service Robotics* 17.5 (2024), pp. 1091–1107 (cit. on pp. 4, 12).

[10] Chao Zhang, Zenghui Wang, Guanghui Zhou, et al. „Towards new-generation human-centric smart manufacturing in Industry 5.0: A systematic review“. In: *Advanced Engineering Informatics* 57 (2023), p. 102121 (cit. on pp. 4, 11).

[11] Jiewu Leng, Jiwei Guo, Junxing Xie, et al. „Review of manufacturing system design in the interplay of Industry 4.0 and Industry 5.0 (Part I): Design thinking and modeling methods“. In: *Journal of Manufacturing Systems* 76 (2024), pp. 158–187 (cit. on p. 10).

[12] Huihui Guo, Fan Wu, Yunchuan Qin, et al. „Recent trends in task and motion planning for robotics: A survey“. In: *ACM Computing Surveys* 55.13s (2023), pp. 1–36 (cit. on p. 11).

[13] Zhigen Zhao, Shuo Cheng, Yan Ding, et al. „A survey of optimization-based task and motion planning: From classical to learning approaches“. In: *IEEE/ASME Transactions on Mechatronics* (2024) (cit. on pp. 11, 13, 15).

[14] Santeri Vallin. „Integrating Large Language Models into PDDL-Based Robot Task and Motion Planning“. In: (2024) (cit. on p. 11).

[15] Iakovos T Michailidis, Panagiotis Michailidis, Elias Kosmatopoulos, and Nick Bassiliades. „Open-Source Online Mission-Planning in Emergent Environments with PDDL for Multi-robot Applications“. In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2024, pp. 433–446 (cit. on p. 11).

[16] Guang Hu, Tim Miller, and Nir Lipovetzky. „Planning with Perspectives–Decomposing Epistemic Planning using Functional STRIPS“. In: *Journal of Artificial Intelligence Research* 75 (2022), pp. 489–539 (cit. on p. 11).

[17] Qiguang Chen and Ya-Jun Pan. „An optimal task planning and agent-aware allocation algorithm in collaborative tasks combining with pddl and popf“. In: *arXiv preprint arXiv:2407.08534* (2024) (cit. on p. 11).

[18] Chaoyang Zhu. „Intelligent robot path planning and navigation based on reinforcement learning and adaptive control“. In: *J. Logist. Inform. Serv. Sci* 10.3 (2023), pp. 235–248 (cit. on p. 12).

[19] Kuo-Ching Ying, Pourya Pourhejazy, Chen-Yang Cheng, and Zong-Ying Cai. „Deep learning-based optimization for motion planning of dual-arm assembly robots“. In: *Computers & Industrial Engineering* 160 (2021), p. 107603 (cit. on p. 12).

[20] Runqing Miao, Qingxuan Jia, and Fuchun Sun. „Long-term robot manipulation task planning with scene graph and semantic knowledge“. In: *Robotic Intelligence and Automation* 43.1 (2023), pp. 12–22 (cit. on p. 12).

[21] Tom Brown, Benjamin Mann, Nick Ryder, et al. „Language models are few-shot learners“. In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901 (cit. on p. 12).

[22] Josh Achiam, Steven Adler, Sandhini Agarwal, et al. „Gpt-4 technical report“. In: *arXiv preprint arXiv:2303.08774* (2023) (cit. on p. 12).

[23] Jacob Devlin. „Bert: Pre-training of deep bidirectional transformers for language understanding“. In: *arXiv preprint arXiv:1810.04805* (2018) (cit. on p. 12).

[24] Colin Raffel, Noam Shazeer, Adam Roberts, et al. „Exploring the limits of transfer learning with a unified text-to-text transformer“. In: *Journal of machine learning research* 21.140 (2020), pp. 1–67 (cit. on p. 12).

[25] Naveen Arivazhagan, Ankur Bapna, Orhan Firat, et al. „Massively multilingual neural machine translation in the wild: Findings and challenges“. In: *arXiv preprint arXiv:1907.05019* (2019) (cit. on p. 12).

[26] Raymond Li, Loubna Ben Allal, Yangtian Zi, et al. „Starcoder: may the source be with you!“ In: *arXiv preprint arXiv:2305.06161* (2023) (cit. on p. 12).

[27] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated Planning: theory and practice*. Elsevier, 2004 (cit. on p. 13).

[28] Marcel Steinmetz. „Conflict-driven learning in AI planning state-space search“. In: (2022) (cit. on p. 13).

[29] Nikolaus Frohner. „Advancing state space search for static and dynamic Optimization by parallelization and learning“. PhD thesis. Technische Universität Wien, 2023 (cit. on p. 13).

[30] Silvia Izquierdo-Badiola, Gerard Canal, Guillem Alenyà, Carlos Rizzo, and Andrew Coles. „Planning for Human-Robot Collaboration Scenarios with Heterogeneous Costs and Durations“. In: *ECAI 2024*. 2024 (cit. on p. 13).

[31] Dancheng Gao, Andrew Coles, and Amanda Coles. „Learning Macro-Actions to Improve the Relaxed Planning Graph Heuristic“. In: () (cit. on p. 13).

[32] Viraj Parimi, Zachary B Rubinstein, and Stephen F Smith. „T-htn: Timeline based htn planning for multiple robots“. In: *Proceedings of the 5th ICAPS Workshop on Hierarchical Planning 32nd International Conference on Automated Planning and Scheduling*. 2022, pp. 59–67 (cit. on p. 14).

[33] Ebaa Alnazer, Ilche Georgievski, and Marco Aiello. „Risk Awareness in HTN Planning“. In: *arXiv preprint arXiv:2204.10669* (2022) (cit. on p. 14).

[34] Rohit Singh and Indranil Saha. „An Online Planning Framework for Multi-Robot Systems with LTL Specification“. In: *2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE. 2024, pp. 180–191 (cit. on p. 14).

[35] Harsha Kokel, Nikhilesh Prabhakar, Balaraman Ravindran, et al. „Hybrid deep reprel: Integrating relational planning and reinforcement learning for information fusion“. In: *2022 25th International Conference on Information Fusion (FUSION)*. IEEE. 2022, pp. 1–8 (cit. on p. 15).

[36] Jérémy Turi. „Planning from operational models for deliberate acting in Robotics“. PhD thesis. INSA de Toulouse, 2024 (cit. on p. 15).

[37] Haizhen Li and Xilun Ding. „Adaptive and intelligent robot task planning for home service: A review“. In: *Engineering Applications of Artificial Intelligence* 117 (2023), p. 105618 (cit. on p. 15).

[38] Deshuai Zheng, Jin Yan, Tao Xue, and Yong Liu. „A knowledge-based task planning approach for robot multi-task manipulation“. In: *Complex & Intelligent Systems* 10.1 (2024), pp. 193–206 (cit. on p. 15).

[39] Jungkyoo Shin, Jieun Han, SeungJun Kim, Yoonseon Oh, and Eunwoo Kim. „Task Planning for Long-Horizon Cooking Tasks Based on Large Language Models“. In: *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2024, pp. 13613–13619 (cit. on p. 15).

[40] Yuqian Jiang, Fangkai Yang, Shiqi Zhang, and Peter Stone. „Task-motion planning with reinforcement learning for adaptable mobile service robots“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7529–7534 (cit. on p. 15).

[41] Runqi Chai, Hanlin Niu, Joaquin Carrasco, et al. „Design and experimental validation of deep reinforcement learning-based fast trajectory planning and control for mobile robot in unknown environment“. In: *IEEE Transactions on Neural Networks and Learning Systems* 35.4 (2022), pp. 5778–5792 (cit. on p. 15).

[42] Yu Li, Kai Cheng, Ruihai Wu, et al. „MobileAfford: mobile robotic manipulation through differentiable affordance learning“. In: *2nd Workshop on Mobile Manipulation and Embodied Intelligence at ICRA 2024*. 2024 (cit. on p. 16).

[43] OpenAI. *ChatGPT*. `https://openai.com/chatgpt`. Accessed: 2024-12-18 (cit. on p. 16).

[44] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, et al. „Palm: Scaling language modeling with pathways“. In: *Journal of Machine Learning Research* 24.240 (2023), pp. 1–113 (cit. on p. 16).

[45] Gemini Team, Rohan Anil, Sebastian Borgeaud, et al. „Gemini: a family of highly capable multimodal models“. In: *arXiv preprint arXiv:2312.11805* (2023) (cit. on p. 16).

[46] Yiheng Liu, Tianle Han, Siyuan Ma, et al. „Summary of chatgpt-related research and perspective towards the future of large language models“. In: *Meta-Radiology* (2023), p. 100017 (cit. on p. 16).

[47] Ce Zhou, Qian Li, Chen Li, et al. „A comprehensive survey on pretrained foundation models: A history from bert to chatgpt“. In: *International Journal of Machine Learning and Cybernetics* (2024), pp. 1–65 (cit. on p. 16).

[48] Zhengliang Liu, Tianyang Zhong, Yiwei Li, et al. „Evaluating large language models for radiology natural language processing“. In: *arXiv preprint arXiv:2307.13693* (2023) (cit. on p. 16).

[49] Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. „Prompting large language models with answer heuristics for knowledge-based visual question answering“. In: *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*. 2023, pp. 14974–14983 (cit. on p. 16).

[50] Hanyao Huang, Ou Zheng, Dongdong Wang, et al. „ChatGPT for shaping the future of dentistry: the potential of multi-modal large language model“. In: *International Journal of Oral Science* 15.1 (2023), p. 29 (cit. on p. 16).

[51] Jiaqi Wang, Zihao Wu, Yiwei Li, et al. „Large language models for robotics: Opportunities, challenges, and perspectives“. In: *arXiv preprint arXiv:2401.04334* (2024) (cit. on pp. 17, 34).

[52] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. „Learning to prompt for vision-language models“. In: *International Journal of Computer Vision* 130.9 (2022), pp. 2337–2348 (cit. on pp. 17, 34).

[53] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. „Minigpt-4: Enhancing vision-language understanding with advanced large language models“. In: *arXiv preprint arXiv:2304.10592* (2023) (cit. on pp. 17, 34).

[54] Callie Y Kim, Christine P Lee, and Bilge Mutlu. „Understanding large-language model (llm)-powered human-robot interaction“. In: *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*. 2024, pp. 371–380 (cit. on pp. 17, 34).

[55] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. „Language models as zero-shot planners: Extracting actionable knowledge for embodied agents“. In: *International conference on machine learning*. PMLR. 2022, pp. 9118–9147 (cit. on pp. 17, 36).

[56] Michael Ahn, Anthony Brohan, Noah Brown, et al. „Do as i can, not as i say: Grounding language in robotic affordances“. In: *arXiv preprint arXiv:2204.01691* (2022) (cit. on pp. 17, 36).

[57] Danny Driess, Fei Xia, Mehdi SM Sajjadi, et al. „Palm-e: An embodied multimodal language model“. In: *arXiv preprint arXiv:2303.03378* (2023) (cit. on pp. 17, 36).

[58] Ishika Singh, Valts Blukis, Arsalan Mousavian, et al. „Progprompt: Generating situated robot task plans using large language models“. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2023, pp. 11523–11530 (cit. on pp. 17, 36).

[59] Ray C Goertz. *Master-slave manipulator*. Vol. 2635. Argonne National Laboratory, 1949 (cit. on p. 18).

[60] Jing Luo, Wei He, and Chenguang Yang. „Combined perception, control, and learning for teleoperation: key technologies, applications, and challenges“. In: *Cognitive Computation and Systems* 2.2 (2020), pp. 33–43 (cit. on pp. 18, 19).

[61] Yuling Li, Kun Liu, Wei He, et al. „Bilateral teleoperation of multiple robots under scheduling communication“. In: *IEEE Transactions on Control Systems Technology* 28.5 (2019), pp. 1770–1784 (cit. on pp. 18, 25).

[62] B Xie, H Liu, R Alghofaili, et al. *A review on virtual reality skill training applications. Frontiers in Virtual Reality*. 2021 (cit. on p. 19).

[63] Yu Lei, Zhi Su, and Chao Cheng. „Virtual reality in human-robot interaction: Challenges and benefits“. In: *Electronic Research Archive* 31.5 (2023), pp. 2374–2408 (cit. on p. 19).

[64] Sihan Huang, Baicun Wang, Xingyu Li, et al. „Industry 5.0 and Society 5.0—Comparison, complementation and co-evolution“. In: *Journal of manufacturing systems* 64 (2022), pp. 424–428 (cit. on p. 20).

[65] Xun Xu, Yuqian Lu, Birgit Vogel-Heuser, and Lihui Wang. „Industry 4.0 and Industry 5.0—Inception, conception and perception“. In: *Journal of manufacturing systems* 61 (2021), pp. 530–535 (cit. on p. 20).

[66] Terrence Fong and Charles Thorpe. „Vehicle teleoperation interfaces“. In: *Autonomous robots* 11 (2001), pp. 9–18 (cit. on p. 20).

[67] Jianhong Cui, Sabri Tosunoglu, Rodney Roberts, Carl Moore, and Daniel W Repperger. „A review of teleoperation system control“. In: *Proceedings of the Florida conference on recent advances in robotics*. Citeseer. 2003, pp. 1–12 (cit. on pp. 20, 21).

[68] S Lichiardopol. „A survey on teleoperation“. In: (2007) (cit. on p. 20).

[69] Pattaraphol Batsomboon, Sabri Tosunoglu, and Daniel W Repperger. „A survey of telesensation and teleoperation technology with virtual reality and force reflection capabilities“. In: *International Journal of Modelling and Simulation* 20.1 (2000), pp. 79–88 (cit. on pp. 21, 49, 55).

[70] Miao Hu, Xianzhuo Luo, Jiawen Chen, et al. „Virtual reality: A survey of enabling technologies and its applications in IoT“. In: *Journal of Network and Computer Applications* 178 (2021), p. 102970 (cit. on p. 22).

[71] David Whitney, Eric Rosen, Elizabeth Phillips, George Konidaris, and Stefanie Tellex. „Comparing robot grasping teleoperation across desktop and virtual reality with ROS reality“. In: *Robotics Research: The 18th International Symposium ISRR*. Springer. 2019, pp. 335–350 (cit. on pp. 23, 25).

[72] Günter Niemeyer, Carsten Preusche, Stefano Stramigioli, and Dongjun Lee. „Telerobotics“. In: *Springer handbook of robotics* (2016), pp. 1085–1108 (cit. on p. 23).

[73] Ryan Scott, Apoorva Kapadia, and Ian Walker. „Intuitive interfaces for teleoperation of continuum robots“. In: *Advances in Human Factors in Robots and Unmanned Systems: Proceedings of the AHFE 2018 International Conference on Human Factors in Robots and Unmanned Systems, July 21-25, 2018, Loews Sapphire Falls Resort at Universal Studios, Orlando, Florida, USA 9*. Springer. 2019, pp. 77–89 (cit. on pp. 24, 27).

[74] Samuel S White, Keion W Bisland, Michael C Collins, and Zhi Li. „Design of a high-level teleoperation interface resilient to the effects of unreliable robot autonomy“. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 11519–11524 (cit. on pp. 24, 28).

[75] Kwun Wang Ng, Robert Mahony, and Darwin Lau. „A dual joystick-trackball interface for accurate and time-efficient teleoperation of cable-driven parallel robots within large workspaces“. In: *Cable-Driven Parallel Robots: Proceedings of the 4th International Conference on Cable-Driven Parallel Robots 4*. Springer. 2019, pp. 391–402 (cit. on p. 24).

[76] Stephen Bier, Rui Li, and Weitian Wang. „A full-dimensional robot teleoperation platform“. In: *2020 11th International Conference on Mechanical and Aerospace Engineering (ICMAE)*. IEEE. 2020, pp. 186–191 (cit. on pp. 24, 27).

[77] Haolin Fei, Shijie Lee, Ziwei Wang, et al. „Seamless robot teleoperation: Intuitive control through hand gestures and neural network decoding“. In: *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE. 2024, pp. 1–6 (cit. on p. 25).

[78] Michael E Walker, Hooman Hedayati, and Daniel Szafir. „Robot teleoperation with augmented reality virtual surrogates“. In: *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE. 2019, pp. 202–210 (cit. on p. 25).

[79] Ivo Dekker, Karel Kellens, and Eric Demeester. „Design and evaluation of an intuitive haptic teleoperation control system for 6-dof industrial manipulators“. In: *Robotics* 12.2 (2023), p. 54 (cit. on p. 25).

[80] Maram Sakr, Waleed Uddin, and HF Machiel Van der Loos. „Orthographic vision-based interface with motion-tracking system for robot arm teleoperation: a comparative study“. In: *Companion of the 2020 ACM/IEEE international conference on human-robot interaction*. 2020, pp. 424–426 (cit. on pp. 25, 28).

[81] Luka Peternel, Cheng Fang, Marco Laghi, et al. „Human arm posture optimisation in bilateral teleoperation through interface reconfiguration“. In: *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2020, pp. 1102–1108 (cit. on p. 25).

[82] Soheil Gholami, Marta Lorenzini, Elena De Momi, and Arash Ajoudani. „Quantitative physical ergonomics assessment of teleoperation interfaces“. In: *IEEE Transactions on Human-Machine Systems* 52.2 (2022), pp. 169–180 (cit. on pp. 25, 28).

[83] Matteo Macchini, Fabrizio Schiano, and Dario Floreano. „Personalized telerobotics by fast machine learning of body-machine interfaces“. In: *IEEE Robotics and Automation Letters* 5.1 (2019), pp. 179–186 (cit. on p. 25).

[84] Mohamed El Beheiry, Sébastien Doutreligne, Clément Caporal, et al. „Virtual reality: beyond visualization“. In: *Journal of molecular biology* 431.7 (2019), pp. 1315–1321 (cit. on pp. 26, 50).

[85] Murphy Wonsick and Taskin Padir. „A systematic review of virtual reality interfaces for controlling and interacting with robots“. In: *Applied Sciences* 10.24 (2020), p. 9051 (cit. on pp. 26, 50).

[86] Weiyong Si, Tianjian Zhong, Ning Wang, and Chenguang Yang. „A multimodal teleoperation interface for human-robot collaboration“. In: *2023 IEEE International Conference on Mechatronics (ICM)*. IEEE. 2023, pp. 1–6 (cit. on p. 26).

[87] Lingxiao Meng, Jiangshan Liu, Wei Chai, Jiankun Wang, and Max Q-H Meng. „Virtual reality based robot teleoperation via human-scene interaction“. In: *Procedia Computer Science* 226 (2023), pp. 141–148 (cit. on p. 26).

[88] Jun Nakanishi, Shunki Itadera, Tadayoshi Aoyama, and Yasuhisa Hasegawa. „Towards the development of an intuitive teleoperation system for human support robot using a VR device“. In: *Advanced Robotics* 34.19 (2020), pp. 1239–1253 (cit. on pp. 26, 29).

[89]Daria Trinitatova and Dzmitry Tsetserukou. „Study of the Effectiveness of a Wearable Haptic Interface With Cutaneous and Vibrotactile Feedback for VR-Based Teleoperation“. In: *IEEE Transactions on Haptics* 16.4 (2023), pp. 463–469 (cit. on p. 26).

[90]Mica R Endsley. „Situation awareness: operationally necessary and scientifically grounded“. In: *Cognition, Technology & Work* 17 (2015), pp. 163–167 (cit. on p. 27).

[91]Daniel Rakita, Bilge Mutlu, and Michael Gleicher. „An autonomous dynamic camera method for effective remote teleoperation“. In: *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*. 2018, pp. 325–333 (cit. on p. 27).

[92]WOJCIECH CIEŚLAK, SEBASTIAN RODYKOW, and DOMINIK BELTER. „Teleoperation of a six-legged walking robot using a hand tracking interface“. In: *Human-Centric Robotics: Proceedings of CLAWAR 2017: 20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines*. World Scientific. 2018, pp. 527–536 (cit. on p. 27).

[93]Daniel J Rea and Stela H Seo. „Still not solved: A call for renewed focus on user-centered teleoperation interfaces“. In: *Frontiers in Robotics and AI* 9 (2022), p. 704225 (cit. on p. 27).

[94]Tsung-Chi Lin, Achyuthan Unni Krishnan, and Zhi Li. „Shared autonomous interface for reducing physical effort in robot teleoperation via human motion mapping“. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020, pp. 9157–9163 (cit. on p. 28).

[95]Francisco Rodríguez-Sedano, Pere Ponsa, Pablo Blanco-Medina, and Luis Miguel Muñoz. „Interface Design of Haptic Feedback on Teleoperated System“. In: *Iberian Robotics conference*. Springer. 2017, pp. 102–113 (cit. on p. 28).

[96]Dong Wei, Bidan Huang, and Qiang Li. „Multi-view merging for robot teleoperation with virtual reality“. In: *IEEE Robotics and Automation Letters* 6.4 (2021), pp. 8537–8544 (cit. on p. 28).

[97]Yi Chen, Baohua Zhang, Jun Zhou, and Kai Wang. „Real-time 3D unstructured environment reconstruction utilizing VR and Kinect-based immersive teleoperation for agricultural field robots“. In: *Computers and Electronics in Agriculture* 175 (2020), p. 105579 (cit. on p. 28).

[98]Abdeldjallil Naceri, Dario Mazzanti, Joao Bimbo, et al. „Towards a virtual reality interface for remote robotic teleoperation“. In: *2019 19th International Conference on Advanced Robotics (ICAR)*. IEEE. 2019, pp. 284–289 (cit. on p. 29).

[99] Yunpeng Su, Xiaoqi Chen, Tony Zhou, Christopher Pretty, and Geoffrey Chase. „Mixed reality-integrated 3D/2D vision mapping for intuitive teleoperation of mobile manipulator“. In: *Robotics and Computer-Integrated Manufacturing* 77 (2022), p. 102332 (cit. on p. 29).

[100] Jeffrey I Lipton, Aidan J Fay, and Daniela Rus. „Baxter's homunculus: Virtual reality spaces for teleoperation in manufacturing“. In: *IEEE Robotics and Automation Letters* 3.1 (2017), pp. 179–186 (cit. on p. 29).

[101] Yulong Li, Shubham Agrawal, Jen-Shuo Liu, Steven K Feiner, and Shuran Song. „Scene editing as teleoperation: A case study in 6dof kit assembly“. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2022, pp. 4773–4780 (cit. on p. 29).

[102] Christian Barentine, Andrew McNay, Ryan Pfaffenbichler, et al. „A vr teleoperation suite with manipulation assist“. In: *Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*. 2021, pp. 442–446 (cit. on p. 30).

[103] Tianyu Zhou, Qi Zhu, and Jing Du. „Intuitive robot teleoperation for civil engineering operations with virtual reality and deep learning scene reconstruction“. In: *Advanced Engineering Informatics* 46 (2020), p. 101170 (cit. on p. 30).

[104] Filippo Brizzi, Lorenzo Peppoloni, Alessandro Graziano, et al. „Effects of augmented reality on the performance of teleoperated industrial assembly tasks in a robotic embodiment“. In: *IEEE Transactions on Human-Machine Systems* 48.2 (2017), pp. 197–206 (cit. on p. 30).

[105] Patrick Stotko, Stefan Krumpen, Max Schwarz, et al. „A VR system for immersive teleoperation and live exploration with a mobile robot“. In: *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 3630–3637 (cit. on p. 30).

[106] David B Van de Merwe, Leendert Van Maanen, Frank B Ter Haar, et al. „Human-robot interaction during virtual reality mediated teleoperation: How environment information affects spatial task performance and operator situation awareness“. In: *Virtual, Augmented and Mixed Reality. Applications and Case Studies: 11th International Conference, VAMR 2019, Held as Part of the 21st HCI International Conference, HCII 2019, Orlando, FL, USA, July 26–31, 2019, Proceedings, Part II 21*. Springer. 2019, pp. 163–177 (cit. on p. 30).

[107] Jin Sol Lee, Youngjib Ham, Hangue Park, and Jeonghee Kim. „Challenges, tasks, and opportunities in teleoperation of excavator toward human-in-the-loop construction automation“. In: *Automation in Construction* 135 (2022), p. 104119 (cit. on p. 49).

[108]J Ernesto Solanes, Adolfo Muñoz, Luis Gracia, and Josep Tornero. „Virtual reality-based interface for advanced assisted mobile robot teleoperation“. In: *Applied Sciences* 12.12 (2022), p. 6071 (cit. on p. 51).

[109]Morgan Quigley, Ken Conley, Brian Gerkey, et al. „ROS: an open-source Robot Operating System“. In: *ICRA workshop on open source software*. Vol. 3. 3.2. Kobe, Japan. 2009, p. 5 (cit. on p. 52).

[110]Lin Zhang, Robert Merrifield, Anton Deguet, and Guang-Zhong Yang. „Powering the world's robots—10 years of ROS“. In: *Science Robotics* 2.11 (2017), eaar1868 (cit. on p. 52).

[111]Unity Technologies. *Unity Real-Time Development Platform | 3D, 2D, VR & AR Engine*. `https://unity.com`. Accessed: 2024-12-18 (cit. on p. 55).

[112]Wei Wang, Yuki Suga, Hiroyasu Iwata, and Shigeki Sugano. „OpenVR: A software tool contributes to research of robotics“. In: *2011 IEEE/SICE International Symposium on System Integration (SII)*. IEEE. 2011, pp. 1043–1048 (cit. on p. 56).

[113]Unity Technologies. *Unity Robotics Hub: Central repository for tools, tutorials, resources, and documentation for robotics simulation in Unity*. `https://github.com/Unity-Technologies/Unity-Robotics-Hub`. Accessed: 2024-12-18 (cit. on pp. 60, 61).

[114]Gigih Priyandoko and MS Hendriyawan Achmad. „Mapping and Navigation for Indoor Robot Using Multiple Sensor Under ROS Framework“. In: *Enabling Industry 4.0 through Advances in Mechatronics: Selected Articles from iM3F 2021, Malaysia*. Springer, 2022, pp. 1–10 (cit. on p. 68).

[115]Sachin Chitta, Ioan Sucan, and Steve Cousins. „Moveit![ros topics]“. In: *IEEE robotics & automation magazine* 19.1 (2012), pp. 18–19 (cit. on pp. 68, 70).

[116]Sachin Chitta. „MoveIt!: an introduction“. In: *Robot Operating System (ROS) The Complete Reference (Volume 1)* (2016), pp. 3–27 (cit. on p. 68).

[117]Ioan A Sucan, Mark Moll, and Lydia E Kavraki. „The open motion planning library“. In: *IEEE Robotics & Automation Magazine* 19.4 (2012), pp. 72–82 (cit. on p. 69).

[118]James J Kuffner and Steven M LaValle. „RRT-connect: An efficient approach to single-query path planning“. In: *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065)*. Vol. 2. IEEE. 2000, pp. 995–1001 (cit. on p. 70).

[119]Jiagui Chen, Yun Zhao, and Xing Xu. „Improved RRT-connect based path planning algorithm for mobile robots“. In: *IEEE Access* 9 (2021), pp. 145988–145999 (cit. on p. 72).

[120] Vladimir Lyandres and S Briskin. „On an approach to moving-average filtering“. In: *Signal processing* 34.2 (1993), pp. 163–178 (cit. on p. 76).

[121] Universal Robots A/S. *UR5e Lightweight, Versatile Cobot*. https://www.universal-robots.com/products/ur5e/. Accessed: 2024-12-18 (cit. on p. 79).

[122] HTC Corporation. *VIVE Pro Full Kit: The Professional-Grade VR Headset*. https://www.vive.com/hk/product/vive-pro-full-kit/. Accessed: 2024-12-18 (cit. on p. 79).

[123] Microsoft Corporation. *Azure Kinect DK – Develop AI Models | Microsoft Azure*. https://azure.microsoft.com/en-us/products/kinect-dk/. Accessed: 2024-12-18 (cit. on p. 79).