INTERACTION AND CONTROL IN ROBOTICS: FROM ASSISTANCE TO COLLABORATIVE MANIPULATION

LEE HOI YIN

PhD

The Hong Kong Polytechnic University

2024

The Hong Kong Polytechnic University

Department of Mechanical Engineering

Interaction and Control in Robotics: From Assistance to Collaborative Manipulation

Lee Hoi Yin

A thesis submitted in partial fulfilment of
the requirements for the degree of
DOCTOR OF PHILOSOPHY

June 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 acknowledgment has been made in the text.

Signature: __________________________

Name: LEE Hoi Yin

Abstract

Throughout human history, we have collaborated with one another to accomplish more challenging tasks together. With technological advancements, humans have invented a myriad of machines and robots to assist in our daily lives and tasks. Nowadays, we are accustomed to working alongside robots, such as adopting robotic arms to move large metal pieces in car manufacturing.

Beyond using robots to assist humans, researchers have also focused on enabling robots to collaborate and work with one another, introducing multi-robot systems. To further enhance the intelligence of robots and minimize the differences between human and machine, researchers have also developed robots that can understand and use human tools to manipulate objects while carrying out tasks.

This thesis focuses on the interactions and controls in robotics, exploring various aspects from perception and capability sharing in multi-robot collaboration to collaborative manipulation through tool usage. To investigate potential application scenarios, the work includes three case studies: (1) Welder training assistant with augmented perception, (2) Capability sharing in heterogeneous multi-robot systems, and (3) Non-prehensile tool manipulation.

In the welder training assistant project, a multi-sensor interface was developed to assist humans in teaching and learning arc welding more efficiently through performance visualization and quantification. For the multi-robot system, a distributed ontological collaborative task allocation framework was proposed, focusing on allocating tasks among robots based on their capabilities. In the tool manipulation project, a non-prehensile tool manipulation methodology was developed, utilizing a Large Language Model for task decomposition. To enable fine tool motion correction with objects confined in a limited area, an incremental stepping manipulation approach was also designed.

The proposed methodologies are validated and analyzed through extensive experiments to demonstrate the efficiency and effectiveness of the developed solutions in real-world scenarios.

Acknowledgment

I am deeply grateful to my supervisor, Dr. David Navarro-Alarcon, Department of Mechanical Engineering, the Hong Kong Polytechnic University, for his unwavering support and guidance throughout this research journey. His vast knowledge, keen insights, and endless encouragement have been truly invaluable to me. Whenever I encountered challenges, he was there to offer his wisdom and perspective, helping me push through and reach new heights. It has been an absolute honor and privilege to have him as my mentor, and I owe so much of my success to his dedication and belief in me. I feel incredibly fortunate to have had the opportunity to learn from and work alongside such an esteemed and caring professor. His mentorship has not only shaped the outcome of this work, but has also profoundly impacted my growth as a researcher and scholar. I will forever be thankful for his steadfast commitment to my development and the indelible mark he has left on my academic journey.

The contribution from the Department of Mechanical Engineering, the Hong Kong Polytechnic University is genuinely appreciated. Without the support, this research could not have achieved the current results.

I am grateful to all the members of the Robotics and Machine Intelligence Laboratory (ROMI Lab). Their hard work and helpful ideas have been very important for this research. The friendly and collaborative environment in the lab has been crucial for making my academic experience rewarding. I appreciate the support and friendship of each member - without them, I could not have completed this work. The lab has felt like a second home, and I will always remember the time I spent here.

Contents

List of Figures

Chapter 1

Introduction

The term ”robot” was first introduced to the general public in the 1920 Czech play “Rossum’s Universal Robots” (R.U.R.), where it was used to describe artificial humanoid machines designed for ”forced labor.” Since then, the concept of robots has evolved significantly. With our modern robotics development, there are a wide range of advanced machines and systems that have become deeply integrated into our lives.

Today, robots are employed in diverse applications, from industrial manufacturing to healthcare, service, and beyond. With the recent rapid development and breakthroughs in fields like hardware engineering, electronics, computing, and artificial intelligence, the capabilities of a robot expands significantly, influencing not only the researchers but also with the general public. Nowadays, robot collaborating with humans or even autonomously completing complex tasks with collaborative tool manipulation are no longer the privileges of roboticist. With the great breakthrough in generative artificial intelligence, robot can now understand human natural language, and interpret with various sensors input, to generate ra-

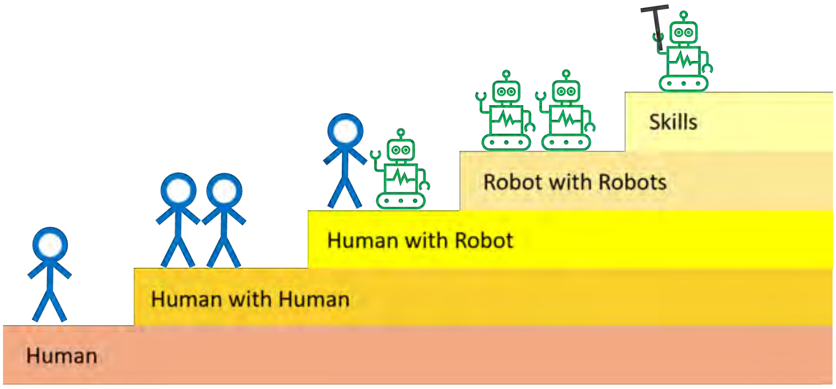


Figure 1.1: Collaboration evolution history, from human collaborates with human to multi-robot collaboration. The focus of this thesis starts from robot assists human, capability sharing in robots collaboration, to the skills enhancement of a robot.

tional motion plans for autonomous execution.

In this thesis, the main research question is “Can we make the robot smarter?”. In more specific, we have raise the follow four sub-questions:

- How can we apply the augmented perception information to robots to assist humans?
- Can we enhance the robot’s capabilities?
- Can we make the robots collaborate?
- Can the robot use tools like us to manipulate objects?

Given the above questions, the main objectives of this study is to develop solutions to enhance the robot’s capability to reduce the human and robot’s intelligence differences and to validate the proposed methods with real world experiments.

We focus on the robot assistant, multi-robot collaboration, and the skills enhancement on robot as shown in 1.1. We explore the interaction and control aspects of robotics, mainly from perception to collaboration manipulation. Three key research projects are investigated: a welding training platform that leverages robot assistance with argumented perception, the capability sharing collaboration of heterogeneous multi-robot systems, and the capability enhancement with the non-prehensile tool-object manipulation in a dual-arm robotic platforms. These projects aim to push the boundaries of capability in a single body, from assistance to enhancement, contributing to the ongoing advancements in the field of interactive and intelligent robotics.

1.1 Capability Assistance: Human Robot Collaboration

In terms of assistance, our human vision system (i.e. eye) provides the richest information about the environment to our brain. Therefore, we focus on using additional perception information from the bot to assist human. In arc welding training, traditional methods to acquire hand-eye coordination skills are typically conducted through one-to-one instruction where trainees must wear protective helmets and conduct several tests. This approach is inefficient as the harmful light emitted from the electric arc impedes the close monitoring of the process; Practitioners can only observe a small bright spot. To tackle these problems, recent training approaches have leveraged virtual reality to safely simulate the process and visualize the geometry of the workpieces. However, the synthetic nature of these types of simulation platforms reduces their effectiveness as they fail to comprise actual welding interactions with the environment, which hinders the trainees' learning process. To provide users with a real welding experience, we have developed a new automatic multi-sensor extended reality platform for arc welding training. Our system is composed of: (1) An HDR camera, monitoring the real welding spot in real-time; (2) A depth sensor, capturing the 3D geometry of the scene; and (3) A head-mounted VR display, visualizing the process safely. Our innovative platform provides users with a "bot trainer", virtual cues of the seam geometry, automatic spot tracking, and performance scores. To validate the platform's feasibility, we conduct extensive experiments with several welding training tasks. We show that compared with the traditional training practice and recent virtual reality approaches, our automated multi-sensor method achieves better performances in terms of accuracy, learning curve, and effectiveness.

1.2 Capability Sharing: Heterogeneous Multi-Robot Collaboration

Beside adopting a perception assistant in the human-robot collaboration, we also extend the assistance of perception capability of a robot to robot-to-robot type assistance. Specifically, sharing the perception capability or other idle capabilities to assist other robots: capability-based heterogeneous multi-robot collaboration. The collaboration of multiple robots is an attractive topic in the community as it brings advantages to solving complex problems over a single robot with limited resources, but the collaboration of heterogeneous robots still remains largely unexplored. Because of that, we propose a capability matching-based distributed task allocation framework for the collaboration of heterogeneous robots. We evaluate the capability of a single robot by an integrated numerical collaboration method.

Then, the proposed framework is applied to optimally allocate the resources and complete the collaboration. We introduce a hardware control scheme with the interaction-oriented dynamic knowledge graph to resolve the conflicts between the hardware control of different robots. Action sequences are produced by task and motion planning during the collaboration of accomplishing the assigned job. We validate the proposed framework by both simulation and experiment where various capability combinations are checked.

1.3 Capability Enhancement: Robot Collaborative Manipulation

While the collaboration of multiple heterogeneous robots can extend capabilities in solving complex problems, robotic systems still fall short of matching the dexterity and versatility of biological entities, such as crows, in wielding and manipulating tools. Beside sharing capability, we focus on enhancing the robot skill, which specifically is to enable the robot to use tool for object manipulation. To further enrich the robot capability, we investigate the use of Large Language Models (LLMs), tool affordances, and object manoeuvrability for non-prehensile tool-based manipulation tasks. Our new method employs LLMs based on scene information and natural language instructions for symbolic task planning. Using a new tool affordance model derived from visual feedback, we develop a manoeuvrability-driven controller to guide the robot's tool utilisation and manipulation actions. The proposed methodology is evaluated with experiments to prove its effectiveness under various manipulation scenarios.

1.4 Contribution of This Work

To contribute to the aforementioned problem, this thesis provides solutions from the perspectives of capability assistance in human-robot system, capability sharing in multi-robot system, and capability enhancement with tool collaborative manipulation in robot. The proposed approaches focus on the topics of 'Interaction' and interaction-oriented Control' methodologies. The main contributions of this thesis can be summarized as follows:

- Capability Assistance: Human-Robot Collaboration
  - To facilitate efficient learning with minimum human instructors in acquiring welding skills with the novel use of extended reality (XR) technology in education. Leveraging sensor fusion, the proposed system provides students with a welding training environment that offers real-time guidance during their hands-on practice, in contrast to conventional synthetic VR-based platforms.
- Capability Sharing: Multi-robot Collaboration
  - This work reports a capability-based task allocation framework for addressing multi-robot collaboration challenges. The framework aims to enhance task efficiency by considering capability sharing among a group of heterogeneous robots, with the novelty lying in the capability matching and sharing mechanisms.
- Capability Enhancement: Robot Collaborative Manipulation

  - To further extend the capabilities of a robot, this thesis focuses on intelligence and dexterity. We first proposes the use of LLMs to interpret task instructions, followed by the development of a novel affordance and maneuverability model for tool usage. To enable object manipulation in confined areas, an incremental stepping approach is applied with the non-prehensile tool-object interaction, which provides incremental accuracy control and make fine tool motion adjustment over the tool-object-environment interaction, in contrast to traditional prehensile motion.

To ensure the feasibility and effectiveness of the proposed methodologies, this thesis implements all the developed approaches on real-world platforms, with their performance rigorously verified through experimental evaluations involving humans, robots, and manipulators.

1.5 Organization

In this work, we present three cases of study with respect to the perception, capability sharing and enhancement:

- A Multi-Sensor Interface to Improve the Learning Experience in Arc Welding Training Tasks;
- A Distributed Dynamic Framework to Allocate Collaborative Tasks Based on Capability Matching in Heterogeneous Multi-Robot Systems
- Non-Prehensile Tool-Object Manipulation by Integrating LLM-Based Planning and Manoeuvrability-Driven Controls

where the interaction and control aspects of the system are demonstrated in detail. In the following, we shall refer to them as the *XR Welding Training Platform*, the *Multi-Robot Collaboration Systems*, and the *Non-Prehensile Tool-Object Manipulation*.

In the study of the XR Welding Training Platform, this thesis thoroughly explores an effective teaching and learning approach, culminating in the development of two educational platforms. These platforms were subsequently implemented in two mechanical engineering courses to validate the proposed teaching and learning methodology. The detailed specifications and evaluation of the educational platforms are reported in Appendix A.

Chapter 2

State-of-the-art Methods

2.1 Capability Assistance: Human-Robot Collaboration

Arc welding is one of the most common material fusing methods in modern manufacturing [1]. In its most basic form, it uses a controllable electric current to melt a joining metal that, once cooled, binds two metallic parts together [2]. Due to its strong and enduring joining properties, welding is used across numerous economically important fields, such as automotive and aerospace industries, shipbuilding, steel construction, oil and gas pipelines, to name a few instances [3]. Despite its widespread use, teaching and learning a proper welding technique has historically presented many challenges to both instructors and trainees [4].

Traditional approaches to acquiring welding skills are typically conducted as follows: The instructor first explains the working principle of the process and demonstrates it with sample welds [5]; Trainees proceed to conduct hands-on

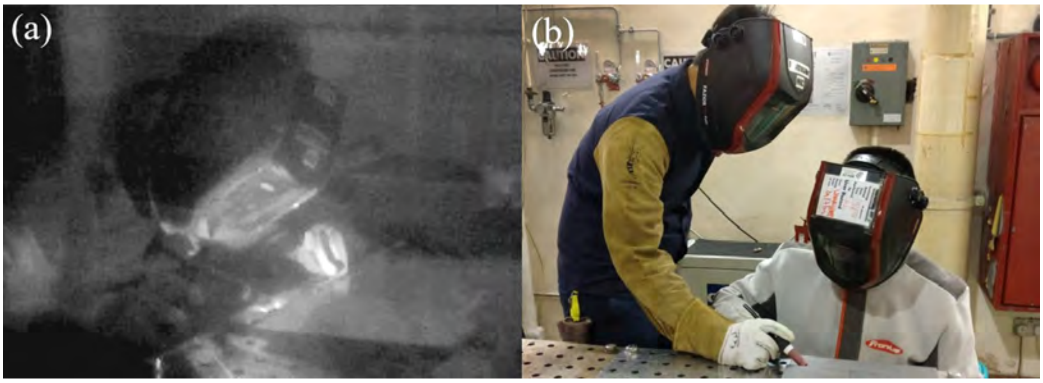


Figure 2.1: (a) Arc welding; (b) Illustration of traditional training approaches, where the trainer needs to demonstrate the welding skill one by one.

welding tasks under the guidance from the instructor, see Fig. 2.1. The main limitation of this approach comes from the inability to clearly observe the workpiece geometry and the process through the protective helmet, which must be worn at all times [6]–[8] (beginners generally struggle to obtain a clear spatial notion of what is occurring on the other side of the helmet). The near-dark experience makes it difficult to learn how the different configurations affect the quality of a weld; It is hard for trainees to understand a process they can barely observe. Furthermore, as there is a limited number of instructors that generally participate in a session, trainees cannot receive real-time support during their practice (advice is received only after the task has been completed and the helmet removed). All these factors complicate the instruction and skill acquisition of arc welding tasks.

2.1.1 Automatic Welding Tracking

During the welding, a strong light generates and hinders the view of the welding seam. To address this issue, the welding seam should be located before the start of the work. In [9], [10], an automatic groove detection and trajectory algorithm

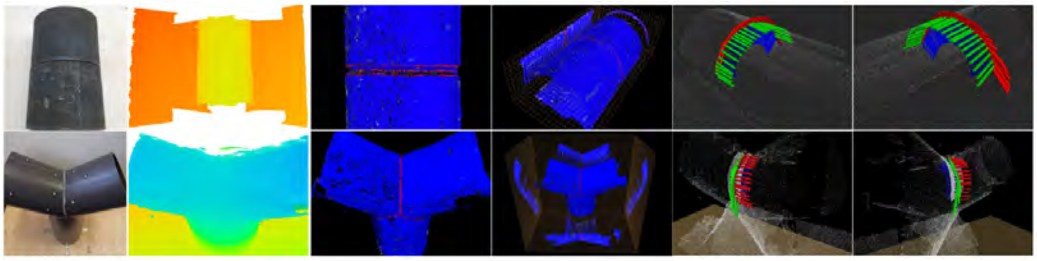
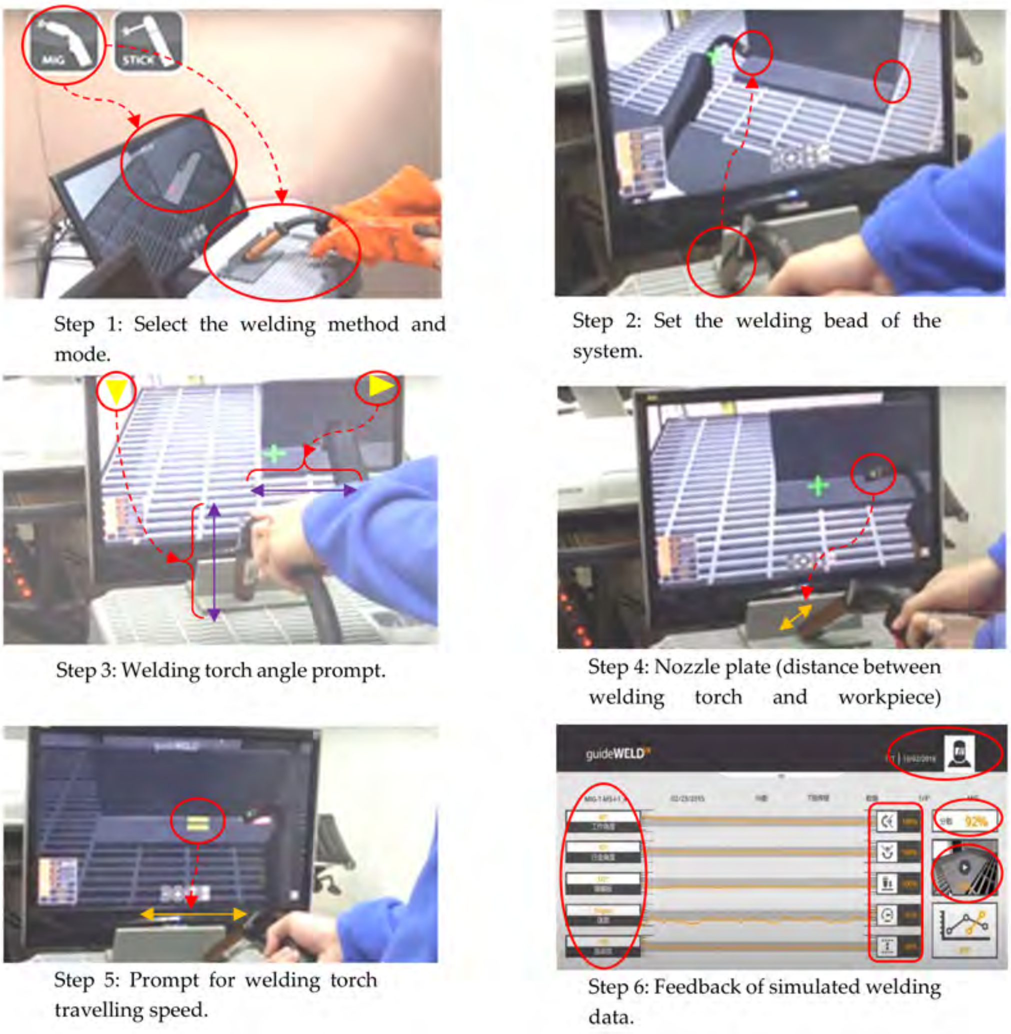


Figure 2.2: Groove detection and trajectory generation by [10].

are presented. The welding groove is located based on the calculation of the point cloud model of the workpiece (see Fig. 2.2). A welding trajectory is computed and updated based on the current information and the past groove data. However, the work is validated with a simulated environment without real-world welding and targets robotic welding. It relies on the RGB camera to update the trajectory calculation during the welding which is infeasible in real welding. Strong light disturbance in the welding process may easily affect the accuracy of the calculation. Human-robot interface or guidance to a worker for training are omitted in the study.

2.1.2 Training System

To address the limitations of traditional training approaches, researchers have developed various didactic platforms based on virtual/mixed reality (VR/MR) systems [11]–[21]. These platforms are typically designed to help beginners to get familiar with the procedure and its torch movements through the use of virtual environments that simulate the *high-energy* welding process (see Fig. 2.3). As no actual interaction with the environment occurs, these simulation-based systems can only provide a synthetic perceptual experience to the users, which may hinder the psychological adaptation that practitioners acquire by conducting real tasks in the field (e.g., fear management, thermal and noise sensations, etc. [22]).



Step 1: Select the welding method and mode.

Step 2: Set the welding bead of the system.

Step 3: Welding torch angle prompt.

Step 4: Nozzle plate (distance between welding torch and workpiece)

Step 5: Prompt for welding torch travelling speed.

Step 6: Feedback of simulated welding data.

Figure 2.3: VR Welding training system developed by [12].

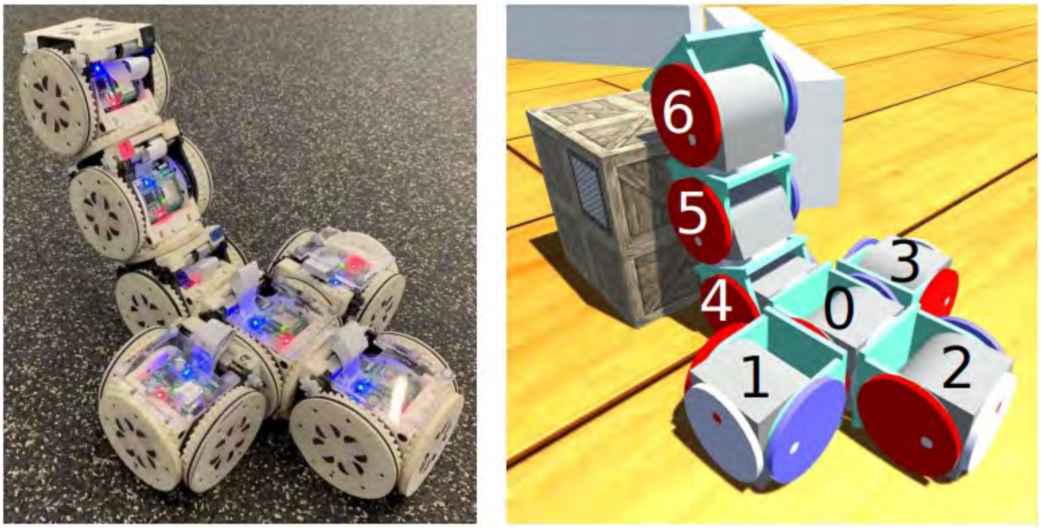


Figure 2.4: A group of self-reconfigurable robot, assembling to form a larger robot as stated in [40].

2.2 Capability Sharing: Heterogeneous Multi-Robot Collaboration

The development of robot fleets that can autonomously assist humans in conducting productive tasks remains an open research problem that demands solutions in the pursuit of building the next generation of smart societies [37]. To reach this ambitious goal, effective collaboration is a key component that can enable groups of heterogeneous robots to improve productivity, as each individual robot can contribute with a specific skill to accomplish the task. Intuitively, a group of five robots (e.g., each with a different carrying capacity, sensing, and locomotion methods) may likely have better efficiency in loading goods than a single robot [38]. This principle of collaboration is naturally adopted by many human/animal groups, however, its effective implementation in multi-robot systems (MRS) is not trivial [39].

Various strategies have been developed to collaboratively perform tasks with MRS, e.g., some representative state-of-the-art methods include [40]–[45]. To optimize the performance of a group of heterogeneous robots, it is essential to develop methods that *characterize* and *exploit* the robots' distinct capabilities in their strategy. Various researchers have addressed this issue [46]–[49], e.g., the approach in [46] presents a capability model and a distributed task allocation algorithm based on auction theory. The suitability of a robot to work on a given task is determined by a metric that quantifies the strength level of the different robots' components. The method in [47] provides a multi-robot collaboration framework that adaptively allocates robots in a task based on their energy consumption. However, most existing approaches only focus on representing robot capabilities for task allocation (i.e., during the planning stage), and do not consider collaborative motions (e.g., forming teams and executing tasks) as part of their capability-based models. This critical issue has not been sufficiently studied in the literature.

Capability *enhancement* is another instrumental aspect of multi-robot collaboration. Robot teams which can reconfigure and augment their functions are expected to be more efficient than teams with a fixed skill set. In recent years, many researchers have developed various self-reconfigurable/self-assemblable systems [40], [42], [50]–[56] (see Fig. 2.4), which have demonstrated high flexibility in their structural properties. The focus of most previous studies has been primarily on hardware modification, however, to algorithmically find an optimal capability enhancement strategy for MRS, we need to develop decision-making methods that incorporate interactions between heterogeneous units with their associated capabilities. Although great progress has been recently achieved, more research needs to be conducted to develop these types of enhancement functionalities.

Table 2.1: Comparison with the features and functions in state-of-the-art methods for MRS.

Methods	Capability	Coll.	Hardware Utilization	Job Execution
[46]	✓	-	-	-
[47]	✓	-	-	✓
[41]	-	✓	-	✓
Ours	✓	✓	✓	✓

The development of *capability-oriented task allocation* strategies is another critical problem. Standard allocation methods have been used by many researchers [57]–[61], who have mostly focused on optimizing the task sequence and execution time, but not on allocating tasks based on the robots' capabilities (which is an under-explored problem). To automatically select different robot types and their roles in the collaborative task, an optimal allocation algorithm may need to consider the semantic relation between the robots, their functions, and the environment. This can be done by using a cognitive architecture graph [62]–[65], a model that can effectively capture this kind of robot-function-environment relationship. Despite its usefulness, cognitive graph methods have not been thoroughly used in task allocation problems involving heterogeneous MRS.

All these previous works have laid very good foundations for the analysis and control of collaborative MRS. However, there are two main (open) problems that hinder the development of task allocation methods for collaborative robots: (1) Difficulty to develop analytical capability models of different robot types in a consistent way; (2) Difficulty to allocate resources/capabilities to execute a collaborative motion task with a team composed of heterogeneous units. The former problem comes mainly due to the lack of a common representation method to characterize the capabilities of robotic systems. Efficient collaboration (e.g., to determine how a specific robot can contribute to the task) requires a shared skill representation framework. The latter problem is due to the fact that traditional approaches typically rely on centralized decision algorithms, which are computationally costly, and unfeasible to implement in heterogeneous systems.

There are many state-of-the-art approaches to facilitate the collaboration between multiple robots. We compare our methods with [41], [46], [47] (see Table 2.1). The method in [46] is a strength-based approach that capability is characterized by its strength level. The decisions calculated by this method are only based on the binary existence of a given skill. Despite that the model in [46], [47] can represent the heterogeneity of the robot's capabilities, these methods cannot capture the merging/augmentation of different robots' capabilities after the assembly. These methods can optimally allocate tasks but do not optimize the utilization of hardware resources. As these methods cannot handle capability augmentation by MRS with heterogeneous capabilities, the available working modes are limited to the traditional approaches (individual or simple group work without optimizing resource utilization). A detailed job execution policy and cognitive knowledge sharing for collaborative works are also not fully studied. The tasks in [41] are divided into sub-tasks where robots make decisions based on Bayesian Delegation and model-based reinforcement learning. Although this method has a more comprehensive framework for collaborative task execution, the model does not consider the robots' capabilities.

2.3 Capability Enhancement: Robot Collaborative Manipulation

Being able to use tools is a widely recognised indicator of intelligence across species [82], [83]. Humans, for instance, have demonstrated mastery of tool use for over two million years [84]. The ability to use tools is invaluable as it extends an organism's reach and enhances its capacity to interact with objects and the environment [82]. Being able to understand the geometric-mechanical relations between tools-objects-environments allows certain species (e.g., apes and crows [85], see Fig. 2.5) to reach food in narrow constrained spaces. For example, a crow can use a stick to drag food out. These pull, drag, and push motions imply the object is moved by a continuous force applied through a tool, without any grasping or seizing involved. The same principles of physical augmentation and its associated non-prehensile manipulation capabilities also apply to robotic systems [86]. For example, by instrumenting them with different types of end-effectors, robots can (in principle) dexterously interact (e.g., push and flip, see Fig. 2.6) with objects of various shapes and masses akin to its biological counterpart [87]–[89]. However, developing this type of manipulation skill is still an open research problem. The goal of Chapter 5 is to develop a methodology to effectively transport objects through non-prehensile tool manipulation actions.

Effective tool utilisation by a robot involves primarily two aspects: (1) task planning and (2) tool movement [90]–[92]. Task planning is typically regarded as a cognitive high-level process in robotics, mainly used for environmental reasoning, task decomposition, allocation of action sequences, etc. [93]. However, recent trends have been pushing towards the use of LLMs to leverage the domain

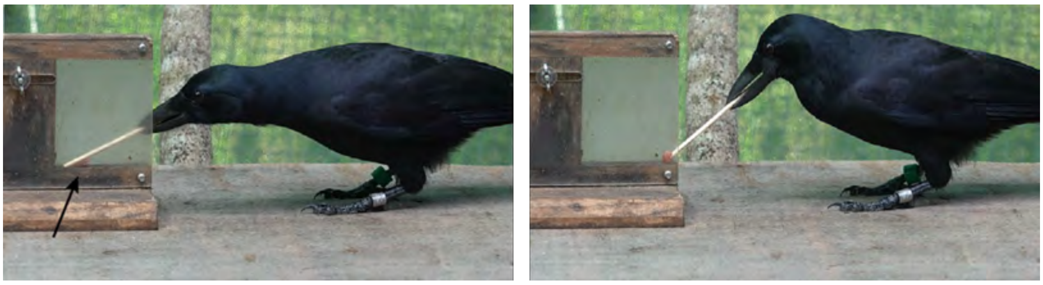


Figure 2.5: A crow using a stick to get the food out from a slot [85].

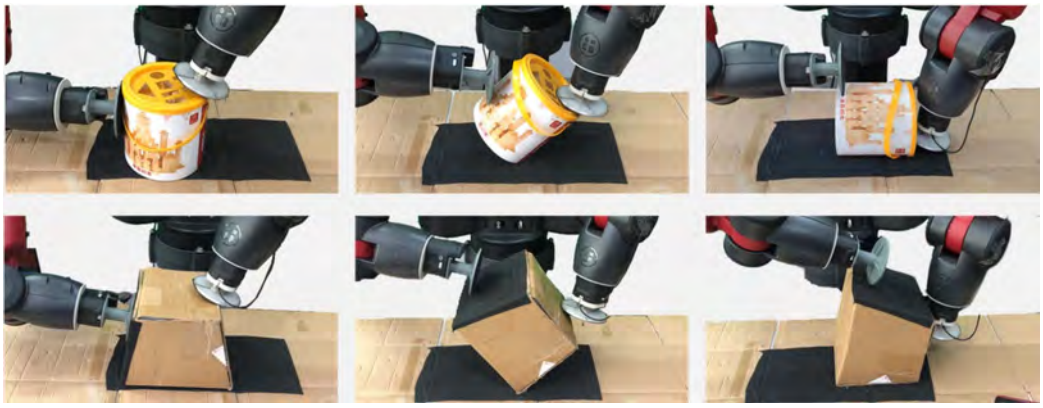


Figure 2.6: A dual arm robot flipping the box into a desired orientation [87].

knowledge for semantically decomposing and planning the execution of manipulation tasks [94]–[97]. Some examples of this directions include [95], [96], which developed an environmental feedback-based system for context-aware improvement planning. Leveraging the generative capabilities of LLMs, motion sequences can be generated for robots as demonstrated in [97]–[99]. The combination of traditional motion planners with LLMs has been explored in [94], [100], [101].

In addition to task planning, various manipulation methodologies have been developed to model the relation between tools and objects [102]. The success of a given tool-object manipulation task largely depends on the appropriate selection of the tool. For example, robots can identify the tool type, potential uses, and

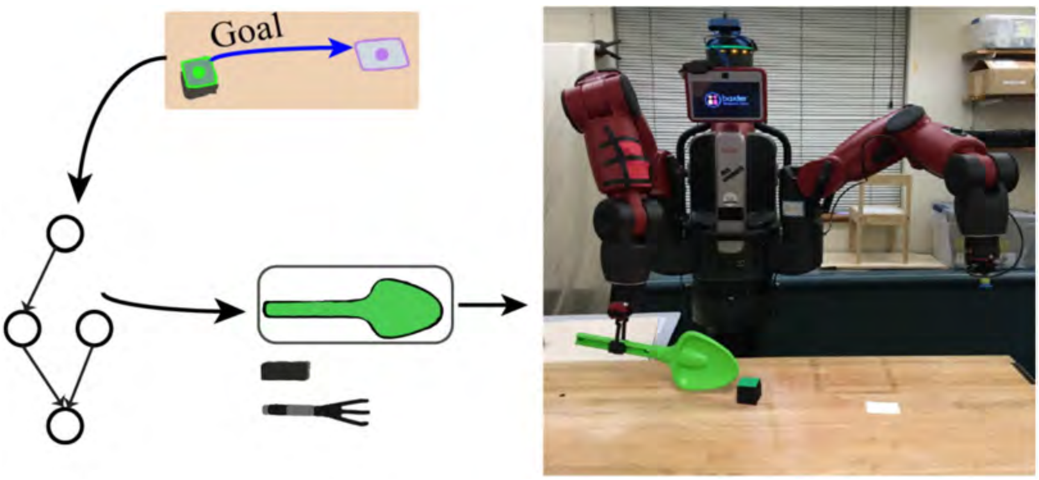


Figure 2.7: A casual approach to obtain the tool affordance feature [104].

contact approaches based on the tool's geometry, see e.g., [83], [90]. In [103], tool features are learned through observation of the task's effects and experimental validation of feature hypotheses. Affordance models are a common technique used for tool feature selection [104]–[106] and tool classification [106]–[108], see Fig. 2.7. The relation between tool actions and its effects on objects is explored in [108], [109], where robots acquire affordance knowledge through predefined actions (e.g., pull, push, rotate). Recently, researchers have also explored the use of LLM in accelerating affordance learning in tool manipulation [83]. Some works have studied tool-based manipulation under constraints and from demonstrations [110], [111]. Non-prehensile object manipulation strategies have been used in [112], [113], see Fig. 2.8.

Although there are many studies on robotic tool use, the collaborative tool-based object manipulation by dual-arm systems based on non-prehensile actions remains an underexplored problem.

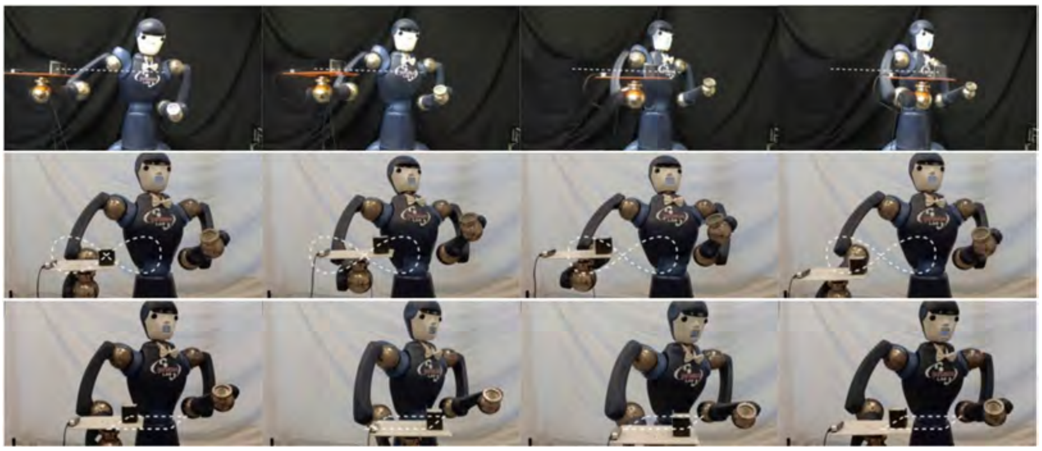


Figure 2.8: A non-prehensile approach is adopted for object balancing in [112].

Chapter 3

Capability Assistance: Human-Robot Collaboration

3.1 Introduction

In most skill learning scenarios, humans typically learn new skills from the demonstrations and guidance of other experienced individuals. This is particularly true in the field of our modern manufacturing, where welding plays a crucial role to join metals [2]. A strong bond requires a good weld, which implies a good welder is needed. In other words, a welder with high quality welding skill is in high demand. In an ideal case, a welder should be trained with one-by-one hands-on practice. However, the number of instructor is in shortage and the eye-hand coordinate skill is hard to acquire in a short time, not to mention the existing welding training platform is a virtual welding machine [12]. To facilitate the efficient teaching and learning of welding, an augmented perception-enabled bot system could serve as an effective assistant to human learners. By providing real-time feedback and guidance based on the learner's actual welding performance, this human-robot collaboration can enable a more effective approach to welder training. Through the integration of advanced perception capabilities, the bot system can monitor the learner's actions, identify areas for improvement, and offer targeted guidance to accelerate the skill development process. By leveraging the complementary strengths of human instructors and intelligent robotic assistants, this approach to welder training can overcome the limitations of existing methods and establish a more efficient and comprehensive skill development pathway. The successful implementation of such a human-robot collaborative framework can have a significant impact on enhancing the quality and accessibility of welder training, ultimately contributing to the advancement of modern manufacturing.

The goal of this real-time extended reality welding training assistance system (refer as "XR bot trainer") is to help trainees to learn the proper torch movements and gain confidence with real welding tasks while reducing the required human instruction to a minimum. We conducted a series of experiments to evaluate our proposed approach in terms of usability, performance, learning curve, and teaching/learning effectiveness.

Compared with existing approaches, the developed automated training system has the following original features:

- It detects and overlays a virtual welding path over the workpiece for trainees to follow.
- It provides the user with instantaneous motion recommendations to improve the task performance.
- It quantifies and visualizes the performance of the conducted welding task

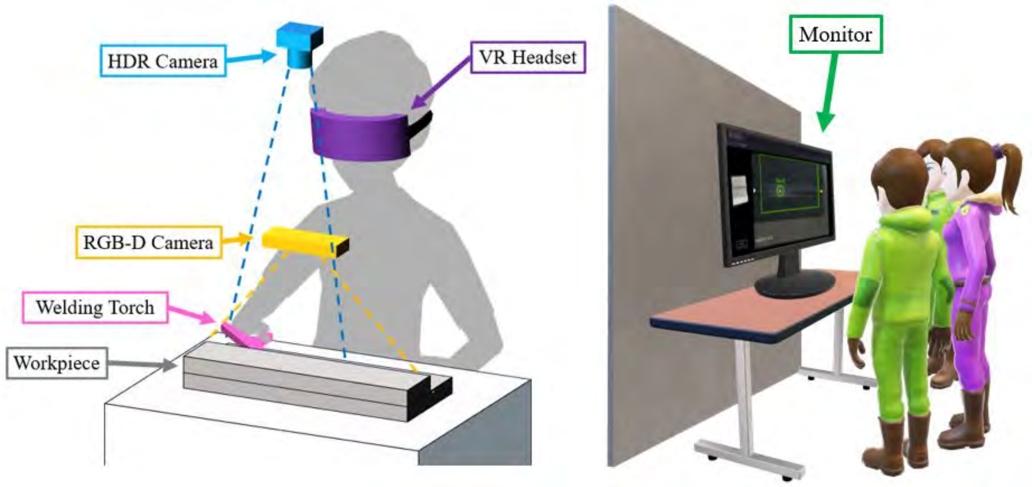


Figure 3.1: Conceptual representation of the proposed training system.

based on sensory feedback.

- It enables to display of the process in real-time to the user and other participants in the training session.

The rest of this chapter is organized as follows: Chapter 3.2 introduces the architecture of the system. Chapter 3.3 presents the experiments. Chapter 3.4 gives the final conclusion.

3.2 Methodology

3.2.1 System Overview

The data flow among the different components of the XR bot trainer is shown in Fig. 3.2. It includes an RGB-D sensor, an HDR camera, a VR headset with a controller, a computer, and a welding torch. Cameras are cross-calibrated and placed at around 10 degrees from the surface normal and facing towards the welding re-

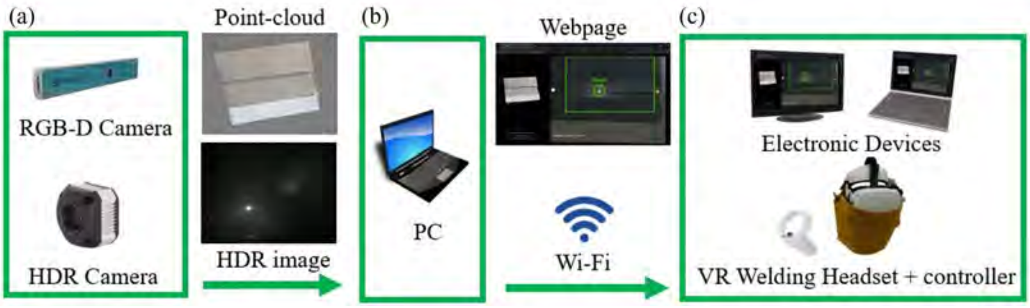


Figure 3.2: The workflow between the different components in the system. (a) Devices involved; (b) PC for image processing; (c) Endpoint devices.

gion. Multimodal visual feedback is sent to the computer for processing and then fed into the VR headset to provide the user with 3D models of the workpiece and a video stream of the welding process.

The workflow depicted in Fig. 3.2 describes the following steps: Before the process begins, (1) the RGB-D sensor captures a depth image of the welding area to produce a point cloud of the workpiece; (2) A depth-based localization algorithm segments the area of interest and computes the welding seam/path [9], [10]. During welding, (3) the HDR camera captures greyscale images of the process and traces the center of the electric arc in real-time, whereas (4) all visual information is registered into a 2D image and streamed to a webpage; (5) The VR headset access the 2D live streaming video and the 3D model via Wi-Fi.

The electrode of the welding torch generates a strong electric arc (whose emitted light is harmful to the human eyes) when approaching an electrified workpiece. The position of this bright spot is tracked by the system, and its trace is visualized to the user to provide valuable visual cues. Upon completion of the task, an evaluation metric is calculated to assess the overall performance; The welding trajectory, average error, and score are displayed on the interface to quantify the

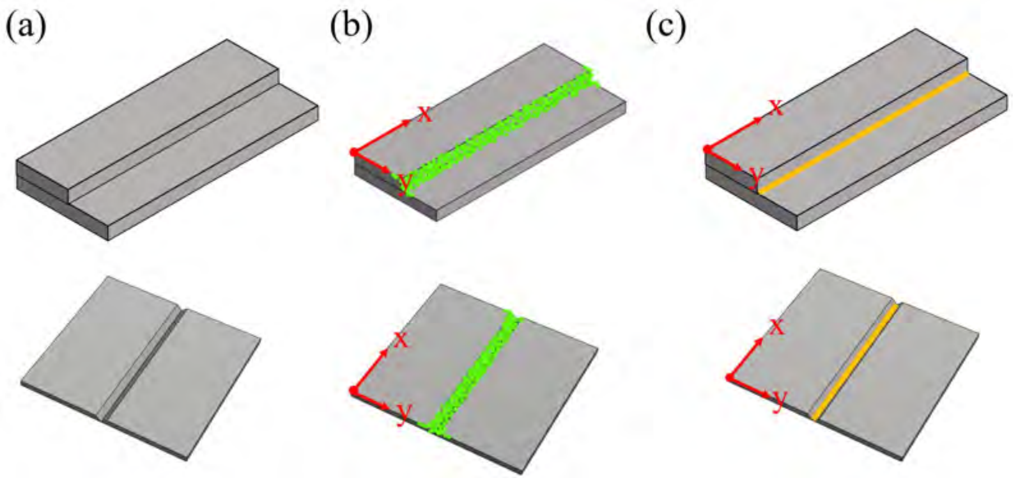


Figure 3.3: (a) Two different types of workpieces: Fillet welding and Butt welding workpieces; (b) The groove detection algorithm is applied to locate the possible welding area. Groove points G are displayed in green color; (c) Welding seams are indicated in orange.

performance of the user.

3.2.2 Seam Localization

A groove is a channel between the edges of two metal workpieces [23]. The RGB-D sensor captures an initial color-depth image that covers the whole welding area of the workpiece; This sensor data forms a point cloud $\mathbf{P}$ of the scene. The proposed detection algorithm uses the difference in edge intensity to automatically find the path of the groove, where local neighborhoods in $\mathbf{P}$ are created to segment the groove's approximated location [9], [10]. In Fig. 3.3, the groove points $G = \{g_1, \ldots, g_n\}$ are conceptually presented in green color over the channel. Computing the bulk of data from this 3D point cloud results in a long processing time; Several points are involved in finding the seam, which (for beginners) is typically a simple straight line. Therefore, the following efficient method for locating the

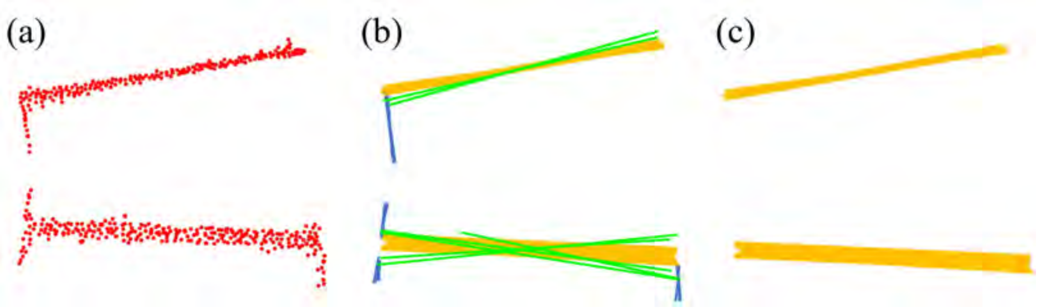


Figure 3.4: (a) 3D coordinates of the groove G are projected onto a 2D image $\mathbf{I}$ in red; (b) The computed lines around the seam are the orange and green long lines; Blue lines represent the side edges of the workpiece, which are generally shorter; (c) Lines in blue and green in (b) are removed due to its length and slope.

seam is implemented: After the groove has been segmented in the image, the seam is calculated by projecting all points in G into a 2D image $\mathbf{I}$, as represented with red color in Fig. 3.4. The groove's edges are then calculated from 2D coordinates.

Noisy data from sensor measurements may affect the precision of the seam localization, thus, a kernel convolution noise filter is implemented on $\mathbf{I}$ to remove noisy/redundant points. Canny edge detection [24] is then applied to $\mathbf{I}$ to sharpen the edges of the groove. Inspired by the method in [25] (which uses an edge map to extract line segments), we compute a point-line connection edge map to improve the precision of our results. Points are interconnected with line segments, as depicted in Fig. 3.4. The endpoints of a line are defined as a and b. Euclidean distance $||a-b||$ is then calculated to determine the length of lines with endpoints a and b. As the welding seam is expected to be longer than the sides of the workpiece, our method removes all short lines.

To extract the characteristic line l representing the seam, we compute the av-

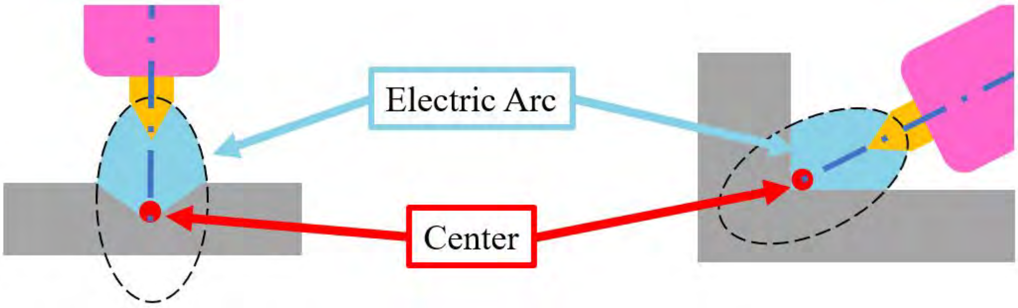


Figure 3.5: Overview of the electric arc and its center during the process.

erage slope of n (long) line segments as:

$$\bar{s} = \frac{1}{n} \sum_{i=1}^{n} s_i \tag{3.1}$$

for s_i as the slope of the ith line segment. Lines with a slope outside a range $[\bar{s} - c, \bar{s} + c]$, for $c > 0$ as clearance scalar, are discarded. This enables us to narrow down the possible seam region to the orange region depicted in Fig. 3.4.

A KD-Tree algorithm [26] is introduced to classify the endpoints $a, b = E(x, y)$ of these lines into groups according to their image quadrants. The two quadrants that contain most of the endpoints indicate the orientation of the seam. The mean value of endpoints in these two quadrants is used to solve the seam characteristic line l. By mapping the depth information in $\mathbf{P}$ related to the points in l, the respective 3D coordinates $\zeta = \{\zeta_1, \ldots, \zeta_n\}$ can be obtained. Segments are assigned to l based on the distance between ζ_1 and ζ_n. A point Q is the target to be followed by the user during the task and is added to visually indicate in the interface's screen.

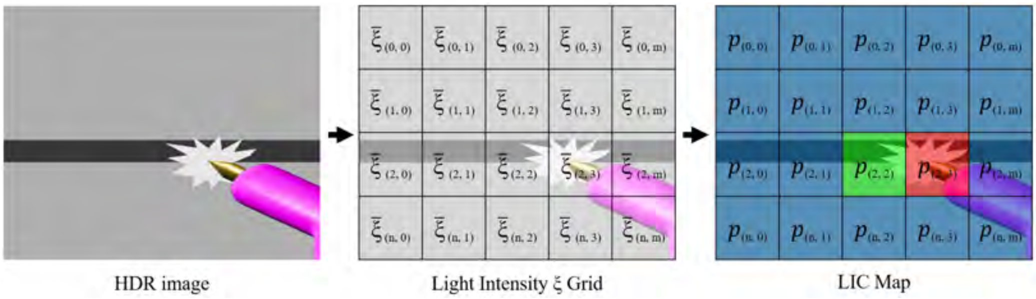

HDR image
$\bar{\xi}_{(0,0)}$ $\bar{\xi}_{(0,1)}$ $\bar{\xi}_{(0,2)}$ $\bar{\xi}_{(0,3)}$ $\bar{\xi}_{(0,m)}$
$\bar{\xi}_{(1,0)}$ $\bar{\xi}_{(1,1)}$ $\bar{\xi}_{(1,2)}$ $\bar{\xi}_{(1,3)}$ $\bar{\xi}_{(1,m)}$
$\bar{\xi}_{(2,0)}$ $\bar{\xi}_{(2,1)}$ $\bar{\xi}_{(2,2)}$ $\bar{\xi}_{(2,3)}$ $\bar{\xi}_{(2,m)}$
$\bar{\xi}_{(n,0)}$ $\bar{\xi}_{(n,1)}$ $\bar{\xi}_{(n,2)}$ $\bar{\xi}_{(n,3)}$ $\bar{\xi}_{(n,m)}$
Light Intensity $\bar{\xi}$ Grid
$p_{(0,0)}$ $p_{(0,1)}$ $p_{(0,2)}$ $p_{(0,3)}$ $p_{(0,m)}$
$p_{(1,0)}$ $p_{(1,1)}$ $p_{(1,2)}$ $p_{(1,3)}$ $p_{(1,m)}$
$p_{(2,0)}$ $p_{(2,1)}$ $p_{(2,2)}$ $p_{(2,3)}$ $p_{(2,m)}$
$p_{(n,0)}$ $p_{(n,1)}$ $p_{(n,2)}$ $p_{(n,3)}$ $p_{(n,m)}$
LIC Map

Figure 3.6: Average light intensity $\bar{\xi}$ is calculated for each grid over the image. An updated LIC $p[t]$ is solved for each grid based on the past confidence level $p[t-1]$ and the current light intensity $\bar{\xi}$ value. Three colors are used to visualize the probability of the potential welding spot location. The red color tile indicates it has the highest confidence level; Green indicates a high confidence level; Blue indicates low confidence.

3.2.3 Electric Arc Localization

Once the welding process starts, a strong light is generated by the electric arc [6], which is perceived through a helmet as a small bright spot; An effective welding task requires moving this spot in the correct direction, i.e., along the seam. By computing the arc's center (shown in Fig. 3.5), the intersection point between the center of the electrode and the seam can be located by our multi-sensor system. As the intensity of the HDR image is affected by the strong light, thus, only the region around the seam is processed and displayed to the user in real-time. Areas far from the seam are statically displayed based on the scene's initial observation (i.e., before the welding process began). The region around the electric arc has high-intensity levels in the captured images, thus, this area of interest is extracted with a binarization approach that generates a feature image. Nevertheless, some light reflections may still appear in the binary map as its computation is susceptible to noises [25].

Light Intensity-Based Confidence Map

To remove flashing noises and minimize the computational time, a confidence map is constructed for each image [27]–[29]. A light intensity-based confidence (LIC) map is applied to the HDR image, where the confidence is determined by the past trajectory and the present observations, see Fig. 3.6. The frame is first divided into multiple tiles, with an average light intensity $\bar{\xi}$ computed for each of them. The light emitted from the welding spot makes the light intensity of the surroundings increase radially from the center of the electric arc. As we consider the slow motions of the welding torch by the user, it is reasonable to assume that for a continuous high intensity in a tile for the past few frames, there is an exponentially high likelihood that an equal or greater light intensity value will be obtained in that same tile in the next frame. Similarly, for a continuous low intensity in the past few frames, it is unlikely there will be an electric arc shown in the coming frames.

Our method aims to reduce the probability density distribution of the whole image. Only high values are considered to create a high contrast image; Thus, an exponential growth/decline is implemented to model the likelihood for the welding spot to be located in a tile of a frame. The confidence level of each grid is then calculated with the following function:

$$p[t] = \min\left(\frac{(b \cdot \mathrm{norm}(\bar{\xi}))^{p[t-1]}}{b}(\sigma + p[t-1]), 1\right) \tag{3.2}$$

where $\mathrm{norm}(\bar{\xi}) = \bar{\xi}/255$ represents the normalized average intensity in each tile; $b > 0$ denotes a user-defined parameter to specify the desired intensity range; $\sigma > 0$ controls the sensitivity to the current observation, where small values result in delayed responses and large values make it susceptible to fast changes (a good compromise is $\sigma = 1$, which is used for the rest of our derivations). The current probability (at time instance t) is denoted by $p[t]$, which describes the confidence of a welding spot to be found in a tile; The method is initialized as $p[t-1] = 0$ for $t = 0$. The minimum value in (3.2) ensures that $0 \leq p[t] \leq 1$ is satisfied.

Different torch poses may affect the light intensity values, making $p[t]$ fluctuate. To achieve stable and consistent values, the confidence is then normalized $p[t]/P_{max}$ based on the maximum $p[t]$ value of the image P_{max}. Using this confidence map, we can estimate the location of the welding spot from the high confidence areas (red regions in Fig. 3.6). We can also predict the next location of the electric arc, labeled in green color in Fig. 3.6.

Dimension Filter

To speed up the computation time, regions extracted with the binarization approach that overlap with a high probability area are only considered for tracking the welding spot. To this end, we first represent these regions by a series contours $K = \{K_1, \ldots, K_n\}$ computed from the binary image. Then, the areas $\alpha = \{\alpha_1, \ldots, \alpha_n\}$ of the contours K are computed by using the Shoelace method $\alpha_i = \frac{1}{2}\,|u - \hat{u}|$, for u and $\hat{u}$ defined as follows [30]:

$$u = q_{nx}q_{iy} + \sum_{i=1}^{n-1} q_{ix}q_{(i+1)y}, \quad \hat{u} = q_{ny}q_{ix} + \sum_{i=1}^{n-1} q_{iy}q_{(i+1)x} \tag{3.3}$$

where $q = \{q_1, \ldots q_m\}$ denotes the set of m image points $q_j = [q_{jx}, q_{jy}]$ of the ith contour K_i. We then evaluate the dimension of the areas α_i and apply the following

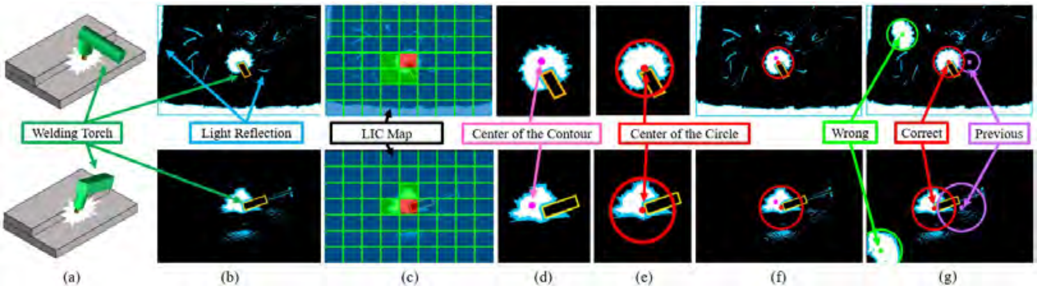


Figure 3.7: (a) The welding torch blocks the view; (b) Some light reflections from the surroundings share a similar light intensity level; (c) A LIC map is generated; (d) Noises in (b) are filtered out; The center of the contour labeled in pink is away from the actual center of the electrode; (e) The correct center can be found with the proposed method; (f) Comparison of the center before and after the use of the method; (g) Demonstration of the verification process.

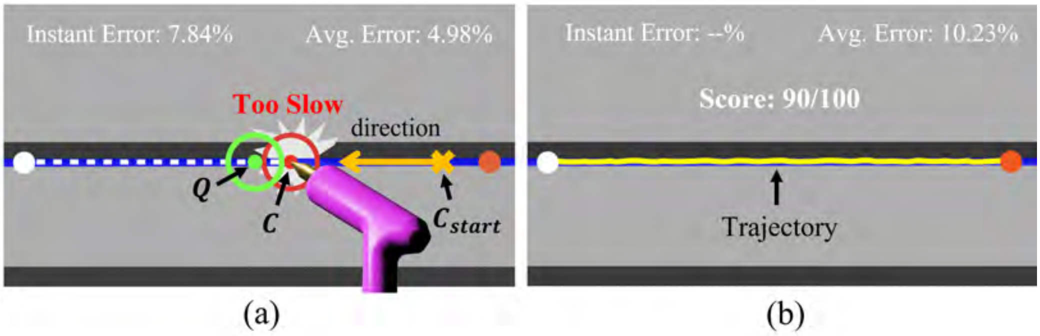


Figure 3.8: The conceptual interface of the XR bot trainer where Q is the recommended point and C is the current welding spot.

condition:

$$\tilde{K}_i = \begin{cases} K_i, & \text{if } A_{min} < \alpha_i < A_{max} \end{cases} \tag{3.4}$$

where $\tilde{K} = \{\tilde{K}_1, \ldots, \tilde{K}_f\}$ denote the *valid* contours that can be used in the tracking algorithm, for $A_{min}, A_{max} > 0$ as scalars to determine this selection.

Electric Arc Tracking

During the task, the user may partially block the view of the electric arc with the torch. This may lead to a shift between the contour's center and the actual center of the arc. To deal with this issue, our method uses a circle that includes all contour points q in α with a minimum radius r_i available. To this end, we model the center of the bounding circle by $C_i = [C_{ix}, C_{iy}]$. As the region that indicates the electric arc generally produces the largest bounding circle, our algorithm verifies these regions starting from the largest circle. However, noises in the image may give rise to the false detection of similar-sized contours, which creates ambiguity for the system.

Algorithm 1 presents the proposed verification process, which is depicted in Fig. 3.7. All valid centers C_i and its radius r_i found from the contour $\tilde{K}_i$ are sorted based on the size of r_i in descending order, i.e., C_1 denotes the center of the largest circle. By comparing the distance between the C_i with the previous state location C_{prev}, the circle can be verified. Through iterating each C_i, the largest reasonable circle around C_{prev} can be located. Kalman filter [31] is applied to further verify the current estimation. The point $C(x,y)$ denotes the detected center of the bounding circle containing the welding spot. Fig. 3.8 conceptually depicts the usage of this point in the proposed interface.

3.2.4 Motion Direction Estimation

The direction of the welding movement depends on the handedness of a user and the location of the seam. For example, a right-hand welder will usually perform right to left motions in a horizontal workpiece; Vertical weld motions are typically

Algorithm 1: Electric Arc Tracking.

```
Input: $C_{prev}$, $\tilde{K}$                    ▷ Previous center, current contours
Output: $C$, $r$                                  ▷ Center and radius of circle
$N \leftarrow$ Compute_Length_of_Array$(\tilde{K})$
for $i \leftarrow 1$ to $N - 1$ do
    $C_i$, $r_i$ = Find_Minimum_Circle($\tilde{K}_i$)
Sort_Cicles_by_Maximum_Radius($C$, $r$)
if $C_{prev}$ is not empty then
    $i \leftarrow 1$
    $valid \leftarrow False$
    while $i \leq N$ & $valid$ is $False$ do
        $valid$ = Compare_Center_Distance($C_i$, $C_{prev}$)
        if $valid$ then
            $C_{prev} \leftarrow C_i$
            return $C_i$, $r_i$
```

from bottom to top. To visually determine the direction of this welding spot, our automated system records the movement of the center C for 20 frames. The starting point C_{start} is computed based on sensory data captured from these first few frames. The moving direction is determined by comparing the signed coordinate difference between the starting point C_{start} and the current point C. For instance, if C is on the left side of C_{start} after 20-frames, it implies that the welder is going from left to right. This way, guidance from the XR bot trainer system can be adjusted accordingly to fit the users' torch motions.

3.2.5 Virtual XR Bot Trainer

The proposed XR bot trainer is responsible for providing instant welding guidance to the user via a head-mounted VR display. The aim of the interface is to provide trainees with the path to be followed by the torch as well as its desired speed. For that, the target point $Q = [Q_x, Q_y]$ is displayed in the VR headset as a moving circle that must be followed by the user by closely placing the electric arc's center C over it. By comparing the coordinates of the target and feedback points, we can compute the *relative* motion of the torch as $\vec{v} = Q - C$, and thus, generate visual cues for the user. For example, if the position and velocity are close to the bot's suggestion, the point Q is displayed with a green circle; Deviations from the target location are displayed with either a red circle or a blue circle, depending on whether C lags or leads the point Q.

The instantaneous position error $\|Q - C\|$ and the average error $\overline{\varepsilon}$ are calculated by the XR bot trainer based on the user's motions; This information is displayed in real-time. After the completion of the task, the system generates a final score and plots the executed welding trajectory. The average error $\overline{\varepsilon}$ is calculated as follows:

$$\overline{\varepsilon} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|C_{xi} - Q_{xi}|}{Q_{xi}} + \frac{|C_{yi} - Q_{yi}|}{Q_{yi}}\right) \tag{3.5}$$

where N is the total number of target points Q involved from C_{start} to C. With this average error, the system then quantifies the task's final score as $\gamma = 100(1 - \overline{\varepsilon})$.

As the proposed method is aimed at lowering the entry barrier for welding, it supports cross-platform usage. The real-time results and the 3D model of the workpiece are available on a webpage, which can be accessed through any electronic device via a wireless connection. This valuable feature enables the instructor and participants to observe the live welding process through a display device.

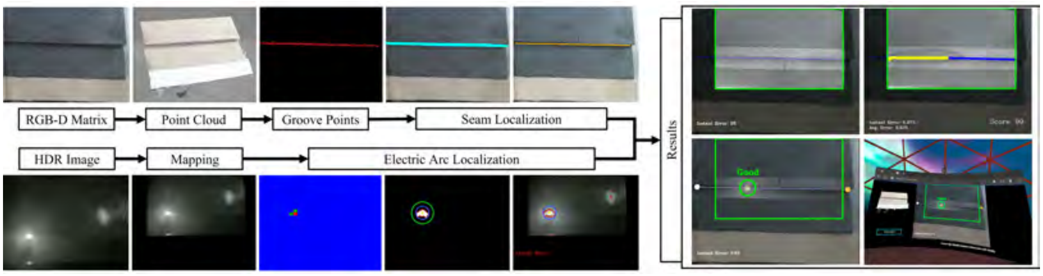


Figure 3.9: Overview of the welding training system. After we place a workpiece, RGB-D and HDR images are captured. The groove detection and the seam localization algorithms are executed to find the welding path. The electric arc localization is applied to the HDR image to compute the center of the arc. The yellow path indicates the welding trajectory with the seam line shown in blue. Images and results are registered into a 2D frame and uploaded to the server. Electronic devices, like a VR headset, are used to access the real-time output via a webpage.

3.3 Results

3.3.1 Experimental Setup

To validate the proposed methodology, a prototype XR interface is built and instrumented with vision sensors to capture the whole welding process. The data processing pipeline is shown in Fig. 3.9, which contains an RGB-D sensor (Intel Realsense SR305), an HDR camera (New Imaging Technology MC1003), a VR headset with a controller (Oculus Quest 2), a PC (GPU RTX 3060), and a standard TIG welding torch. Experiments are conducted to evaluate the system, with two groups of trainees (who have no prior welding experience) being asked to perform several tasks. One group learns with the traditional (i.e. current practice) approach while the other group learns with the proposed multi-sensor interface. In the control group, trainees practice with the instructor and receive one-to-one guidance when necessary; In the experimental group, after the instructor gives a sample demonstration, the trainees practice with the interface and receive no hu-

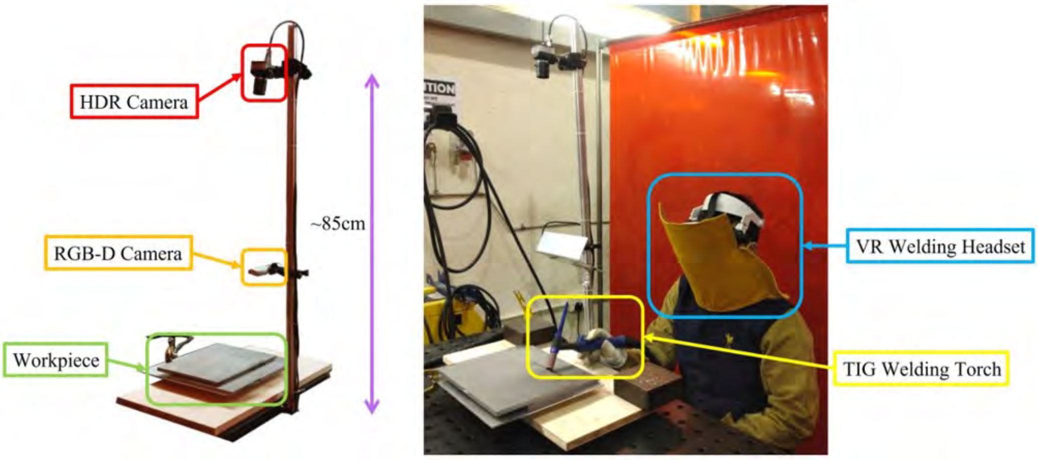


Figure 3.10: Experiment setup of the proposed XR welding training system.

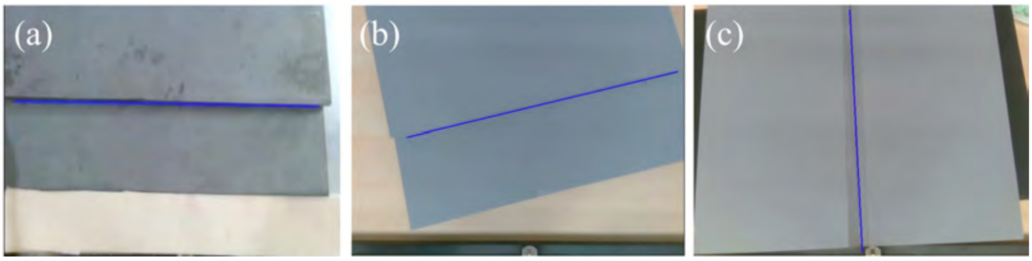


Figure 3.11: Various workpieces, where blue lines indicate the identified seam location. (a)-(b) Fillet workpieces; (c) Butt workpieces.

man guidance. Success and failure are evaluated by the human instructor at the end of each task. Experiments with the interface further evaluate (1) the accuracy of the welding path, (2) the accuracy of the arc location, and (3) the effectiveness of the use of this system in teaching and learning.

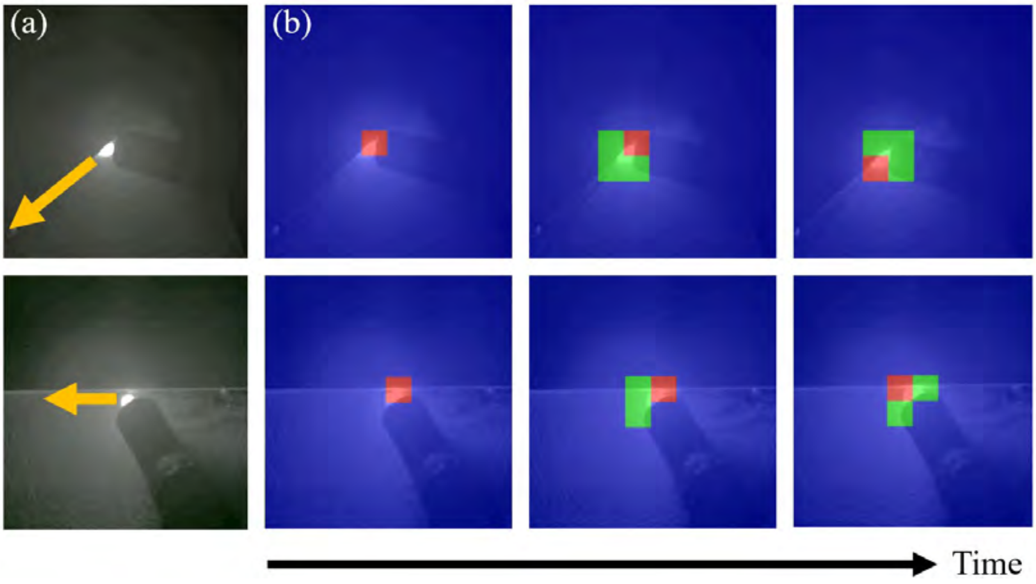


Figure 3.12: Visualization of the welding spot prediction that is generated by the LIC map. (a) The orange arrow indicates the welding direction; (b) Red and Green tiles indicate the potential spot location, from high to medium. Blue indicates low probability areas.

3.3.2 Experimental Design

A horizontal fillet TIG welding is chosen for all experiments, see Fig. 3.10. A non-consumable electrode is used to generate the electric arc. The weld and the molten pool are shielded from environmental contamination by the inert shielding gas [6]. Mild steel workpieces of 10 mm thickness and a straight welding path are used. Although a fillet weld workpiece is prepared, trainees are not required to add any filler at the beginning of their training until they are capable to do so. To obtain a fair comparison between the two methods in terms of learning effectiveness, the task is judged as a success/failure by the human instructor. If

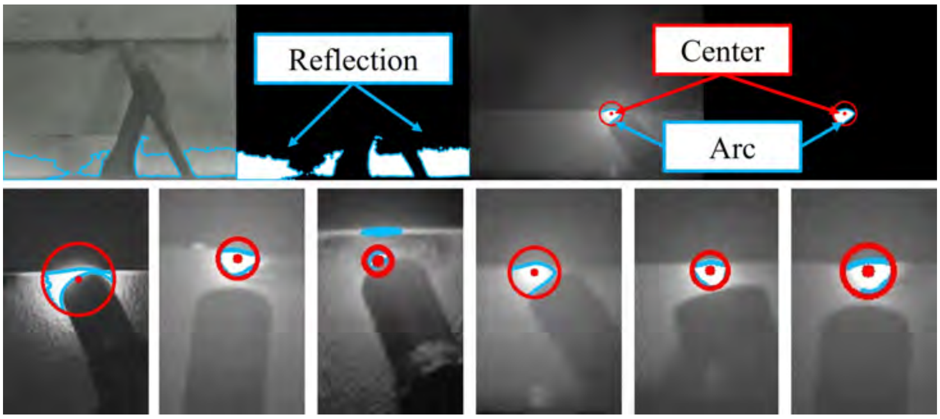


Figure 3.13: Visualization of the contours K, displayed in blue color. Red circles indicate the radius r and the center C.

success is obtained before filler inclusion, then, the trainee can start adding filler in the next trial. If success is reached with filler inclusion, the trainee is considered to have mastered the welding skill and the practice is completed for the trainee.

In the experiments, the system first records the traditional welding process that is conducted by the instructor. Then, the technique is explained to the trainees from these video recordings. Afterward, trainees perform the welding task by using the interface; The instructor and other participants observe the process from a display in *real-time*. After completion, the multimodal recordings can be replayed to analyze various details of the trainees' execution. As proficiency in the task is directly related to the number of repetitions a user has performed [32]–[34], we quantify the system's effectiveness by computing a learning curve that relates the trial number with the obtained score.

3.3.3 Performance Analysis of the Multi-Sensor Interface

As different groove geometries may lead to different seam localization results, we conduct experiments to validate the performance of the automatic seam detection

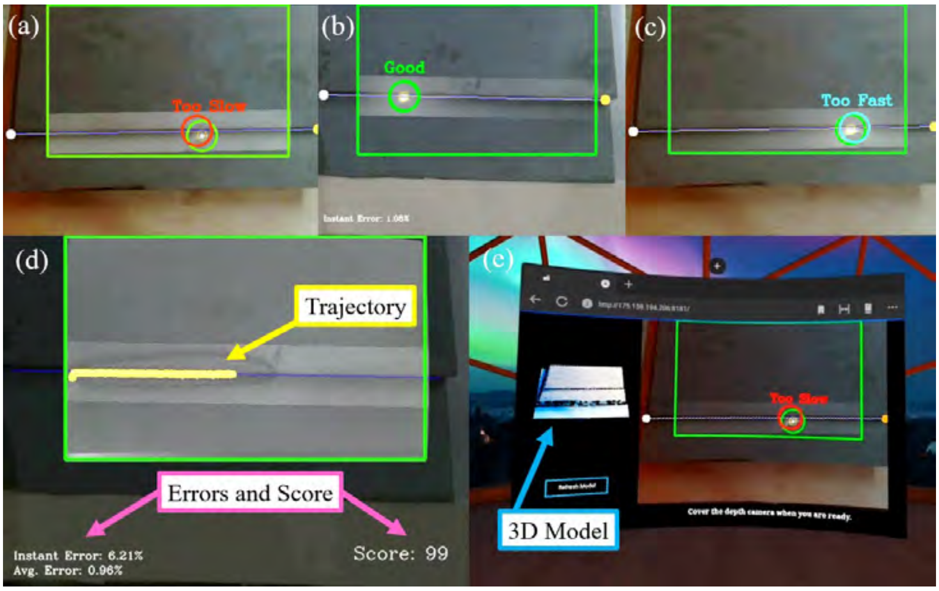
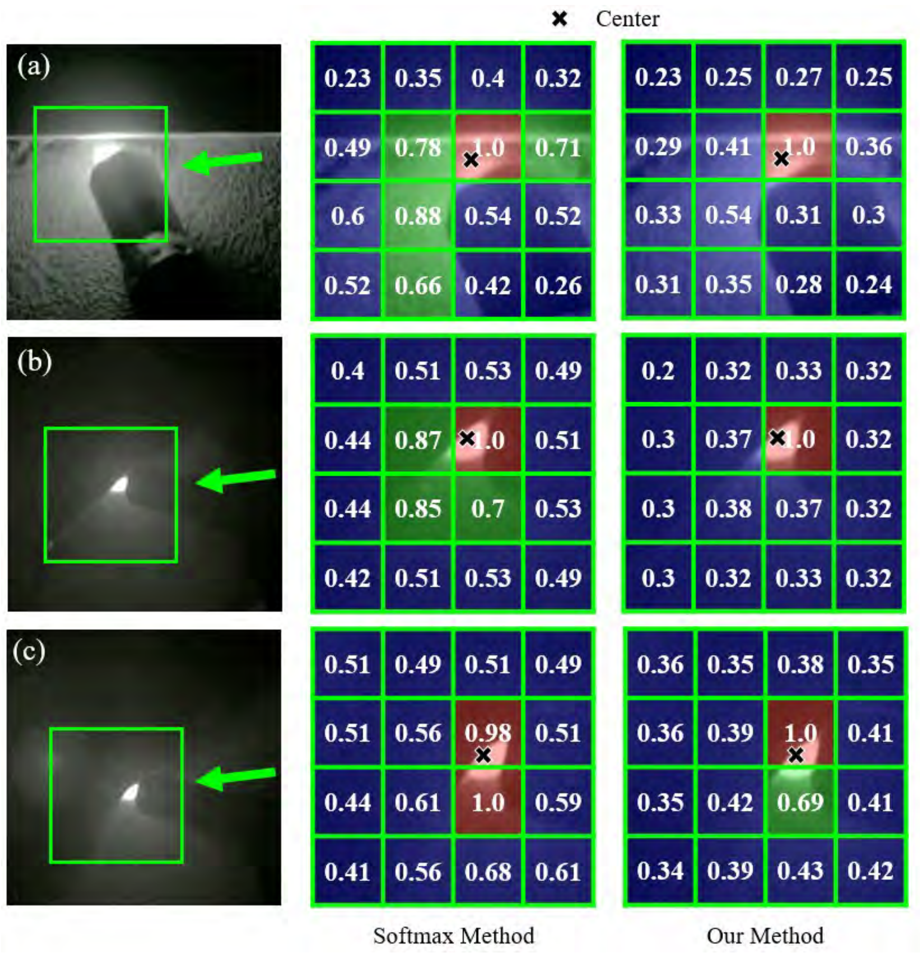


Figure 3.14: (a)-(c) Examples of the welding path and the guidance support provided by the XR bot trainer; The orange and white circles indicate the start and end points of the seam. (d) Welding trajectory, errors, and score of the conducted task; (e) User's view in the VR headset.

algorithm, see Fig. 3.11. Fillet and butt workpieces with various orientations are used for these validation tests, which show that the algorithm can correctly locate different types of seams with various poses. We also conducted experiments to validate the performance of the spot localization and denoising algorithms; We did it in a room with multiple *unknown* light sources to test its robustness. The LIC map is applied to the raw HDR image for noise removal, with $b = 4.5$ in (3.2). We define tiles with $p \geq 0.65$ as areas with high probability, and $p \geq 0.95$ as the estimated welding spot region. These two types of regions are depicted with green and red tiles in Fig. 3.12, where we can see that the LIC map predicts the location of the welding spot from past and current intensity measurements.

In our performance experiments, the welding torch is arranged at various angles and configurations, which sometimes occlude the electric arc. Fig. 3.13 depicts configurations where the welding spot is not entirely captured by the HDR camera. The minimum radius circle r is formed, and thus, the center C can still be computed in these situations. Based on the measured trajectories of the electric arc, the bot assistant provides *real-time* suggestions for the user, as depicted in Fig. 3.14. When the trainee has a slow pace, the suggested target point is displayed in red to alert lagging motion. When the speed of the welding point is too high (which results in insufficient time to melt the material) the circle is displayed in blue. When the welding point is close to the target, the circle is shown in green. These features of our system provide users with valuable guidance to help them improve their technique.



(a) Softmax Method

0.23	0.35	0.4	0.32
0.49	0.78	1.0	0.71
0.6	0.88	0.54	0.52
0.52	0.66	0.42	0.26

(a) Our Method

0.23	0.25	0.27	0.25
0.29	0.41	1.0	0.36
0.33	0.54	0.31	0.3
0.31	0.35	0.28	0.24

(b) Softmax Method

0.4	0.51	0.53	0.49
0.44	0.87	1.0	0.51
0.44	0.85	0.7	0.53
0.42	0.51	0.53	0.49

(b) Our Method

0.2	0.32	0.33	0.32
0.3	0.37	1.0	0.32
0.3	0.38	0.37	0.32
0.3	0.32	0.33	0.32

(c) Softmax Method

0.51	0.49	0.51	0.49
0.51	0.56	0.98	0.51
0.44	0.61	1.0	0.59
0.41	0.56	0.68	0.61

(c) Our Method

0.36	0.35	0.38	0.35
0.36	0.39	1.0	0.41
0.35	0.42	0.69	0.41
0.34	0.39	0.43	0.42

Figure 3.15: Comparison between the proposed LIC Map method and the traditional Softmax method in estimating the electric arc location. The black cross indicates the center of the electric arc. The green tile represents a medium-high belief (≥ 0.65). The red tile represents a high probability value (≥ 0.95). (a)–(b) The confidence value of the surrounding in the Softmax method is higher. (c) A larger error is obtained in estimating the center with the Softmax method.

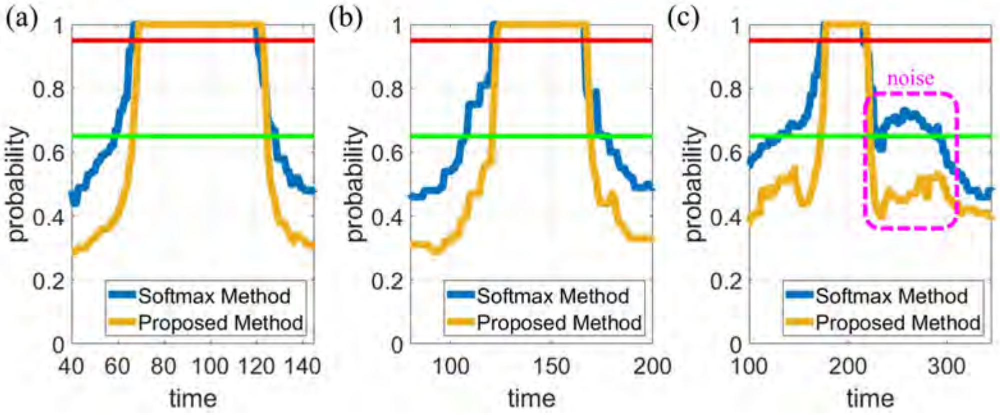


Figure 3.16: Evolution of the probability computed with the LIC Map method and the Softmax method for one tile. The green and red horizontal lines indicate the probability level at 0.65 and 0.95. The pink area is the period when the noise occurs. A steeper slope and a narrower density are obtained by our proposed method. Noise is also greatly suppressed with our exponential decline function.

3.3.4 System Comparison

LIC Map

There are other prediction methods for estimating features similar to the welding spot. The Softmax function estimates the belief based on the current information and the weighted past events [35], [36]. In contrast with this approach, our method (3.2) provides exponential changes in the estimated belief. We compare our method with the classical Softmax [35], [36]:

$$p[t] = \text{norm}(\bar{\xi}) + w \cdot p[t-1] \tag{3.6}$$

where $\text{norm}(\bar{\xi})$ is the normalized light intensity at the current time instance, and $w > 0$ is a scalar weight (which we set to $w = 0.01$). The prediction performance of these methods can be visualized in Fig. 3.15, which shows that the tiles around

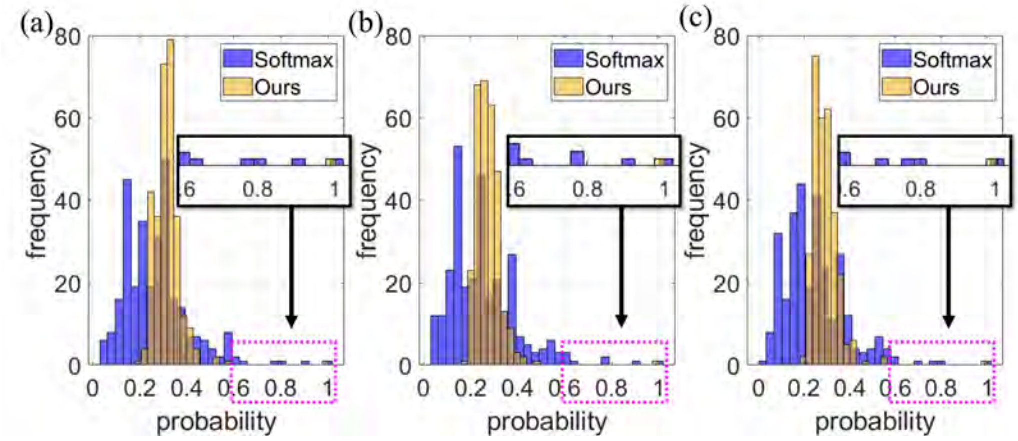


Figure 3.17: Comparison of the histograms obtained with the proposed LIC Map method and the Softmax method. More tiles with ≥ 0.65 probability are found in the Softmax approach, which indicates a higher uncertainty in the estimation.

the welding spot (i.e. the red tile) present a larger (undesired) belief in the Softmax approach than in our method.

To get an insight into the effect of our exponential approach, in Fig. 3.16 we compare the belief changes of a grid when the welding spot passes by it. The figure shows that the confidence level increases and decreases rapidly when the arc approaches the tile; When noises appear, our method can suppress their effect and maintain a continuous low confidence level. The developed LIC map produces a narrow probability distribution (see Fig. 3.17) that results in fewer potential regions for the spot localization (green and red tiles); This leads to a faster denoise process. With our exponential approach, we can sharpen the differences between the useful welding light area and the rest of the image.

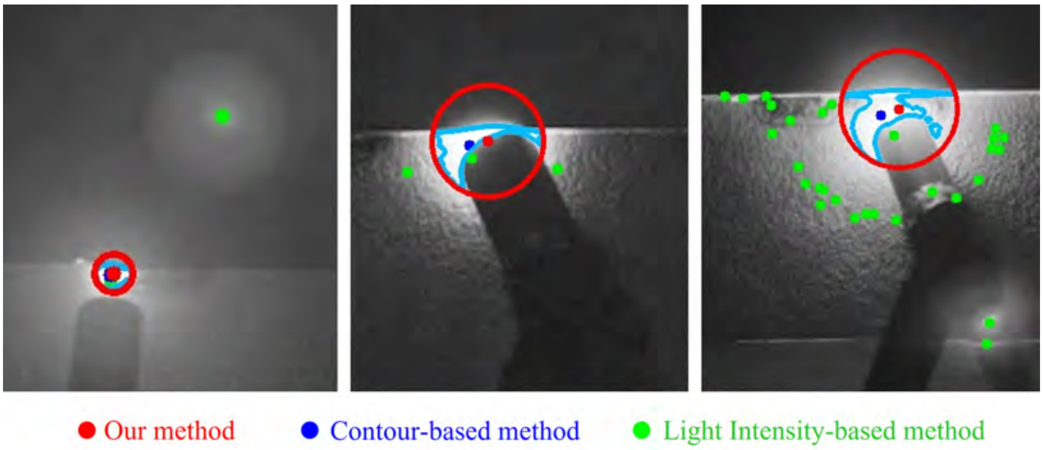


Figure 3.18: Comparison between the proposed method with the state-of-the-art practice in electric arc localization. The red dot is the center determined by our electric arc localization algorithm. The dark blue dot is the center found with a contour-based approach. The green dot is the center identified by the intensity-based approach.

Welding Spot Determination

There are various state-of-the-art approaches to locating the center of the electric arc. Contour-based and light intensity-based methods from the OpenCV library can be used to determine this spot. Fig. 3.18 compares the performance of these methods with our approach. For fairness' sake, the same input data is used for all methods, as a binary image generated by thresholding. In the contour-based method, the center is determined based on the largest contour center in an image. In the light intensity-based approach, the center is found in all the present high light intensity regions without considering the dimension.

The main difference between these three methods is the accuracy in determining the actual welding spot with unpredictable noise inclusion. Noted that the actual welding spot is the center of the electric arc and the ground truth in this comparison is provided by the welder technician. In Fig. 3.18 and Fig. 3.19,

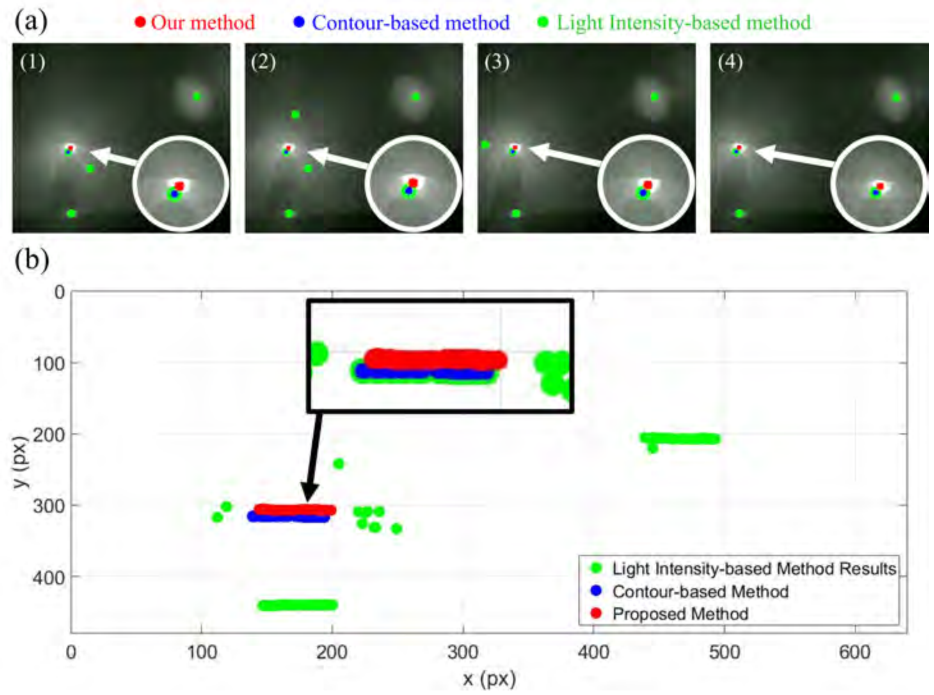


Figure 3.19: Quantitative comparison between the proposed method with the state-of-the-art practice in electric arc localization. (a) The welding process with results overlaid; (b) The position of the electric arc is located by three approaches. Multiple results are obtained from the light intensity-based method.

the contour-based approach misidentifies the starting point of the electric arc as the welding spot. In the light intensity-based approach, more than one "welding spot" is found. With this method, extra image processing for arc localization is normally required. A quantitative comparison between the three methods is presented in Fig. 3.19. A significant error can be observed in the state-of-the-art approaches. This proves that the proposed algorithm can effectively determine the welding spot that is manipulated by the user in noisy environments.

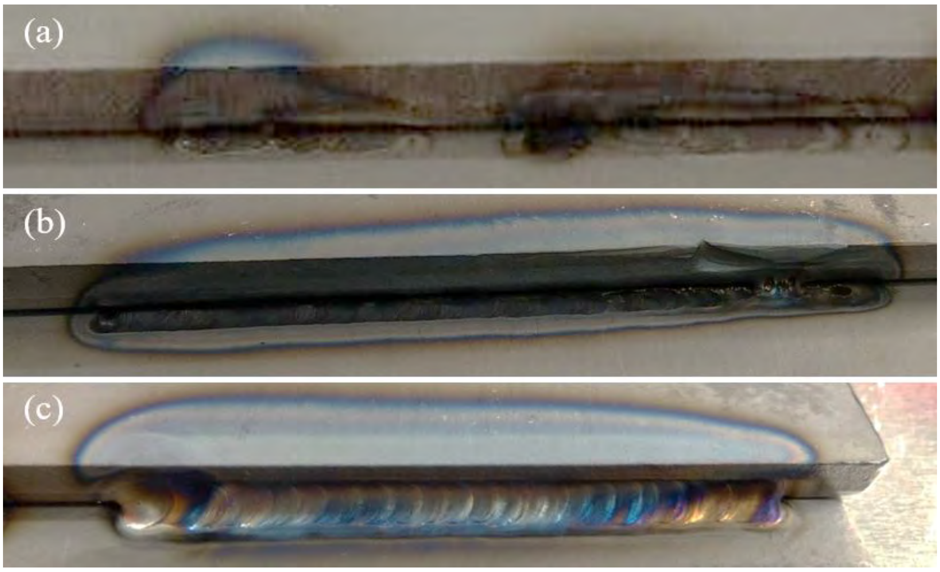


Figure 3.20: Examples of the trainees' performance by independently conducting welding tests. (a) Standard learning approach without filler after 5 trials; (b) Proposed method without filler after 5 trials; (c) Proposed method after 11 trials; (a) Standard learning approach led to consistent failures to maintain a proper movement speed; (b)–(c) Proposed method helped trainees to properly learn the technique, and thus, to master the complete TIG welding skill.

3.3.5 Effectiveness in Teaching and Learning

As the trainees in the *control* group have no prior experience with the XR bot trainer, thus, poor welding performances are observed at first. As they familiarize themselves with the system, it becomes easier to perform the welding task; Filler is successfully added to the weld (a sign of skill mastery) within a short time of hands-on testing, see Fig. 3.20. The performance of the executed welding trajectory is shown in Fig. 3.21. We have evaluated with different welding length and angles. The instant error between the seam and the executed welding trajectory is visualized in Fig. 3.21(b).

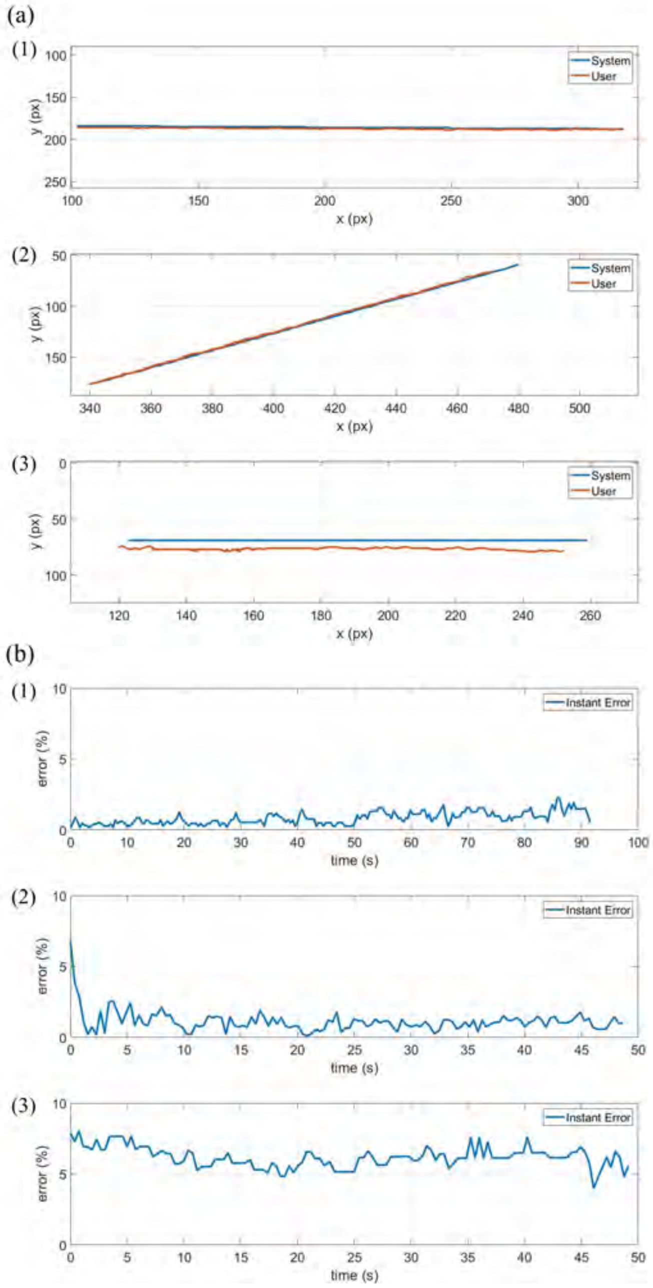


Figure 3.21: Sample welding trajectories performed by the trainees with the aid of our XR system. (a) Spatial x-y motions of the torch, captured during 3 different trials; (b) Instant errors of the trajectories corresponding to the motions in (a).

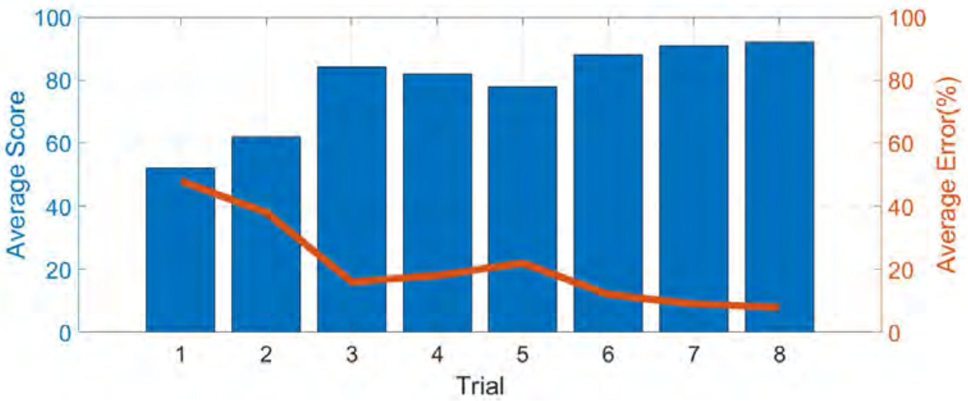


Figure 3.22: Average scores and errors of different welding trials conducted by the group of trainees with our system.

Compared with the standard instructional practice, it is easier for instructors to explain the welding process and techniques to beginners with the proposed interface. In terms of teaching efficiency, the instructor has to keep monitoring and providing guidance to each student one by one in the traditional approach, which is a time-consuming task for them. In the *experimental* group that uses the interface, the instructor is only required to evaluate the task as a success or failure once it is completed. This greatly reduces the instructor's workload since minimal human support is needed. It can be concluded that our XR system can lead to a boost in teaching efficiency. This new approach also enables to increase in the number of trainees that can participate in one session.

With respect to the learning effectiveness, it took around 5 hours for trainees in the control group to master the welding skill while only 3 hours in the experimental group. On average, around 11 trials were needed for a trainee to master the TIG welding skill with our new approach; The multi-sensor interface enabled these participants to *independently* conduct the training tasks. Around 6 more trials were needed when using the traditional approach, with most participants show-

Table 3.1: Comparison with the features and functions in state-of-the-art VR methods for welding training.

	Virtual Welding		Real Welding	
Methods	Seam & Arc Localization	Guidance	Seam & Arc Localization	Guidance
[11]	✓	✓	-	-
[14]	✓	✓	-	-
[15]	✓	✓	-	-
[19]	✓	✓	-	-
Ours	-	-	✓	✓

ing a high dependency on the instructor. The horizontal axis in Fig. 3.22 indicates the number of trials conducted by the trainees in the experimental group. The left vertical axis represents the score obtained during the corresponding trial; The right vertical axis shows the computed average error. As some received success at the eighth trial, no more practices are conducted. These results demonstrate that trainees' proficiency in the task consistently improves with the number of trials. Effectiveness in learning grows with the use of the developed XR bot trainer.

Compared with other VR training studies shown in Table 3.1, our automated XR bot assistant provides immediate *real-time* support to the user, during real-world welding practice. The XR interface provides motion guidance and performance metrics based on sensor feedback. Despite this instant support, our system cannot update the seam's geometric information in real-time since the welding path is computed based on the initial (static) point cloud of the workpiece. However, the instant performance of trainees can be accessed via various types of displays that the instructor can remotely supervise. As virtual welding is only capable of providing a synthetic perceptual experience, trainees cannot address the typical psychological adaptation that they gain through practice in the field.

In the accompanying multimedia file, we demonstrate the performance of the system with multiple experimental videos. https://github.com/romi-lab/VR_Welding/raw/main/video.mp4

3.4 Discussions

This chapter focuses on the capability assistance in human-robot collaboration. Specifically, it presents an XR training bot assistant for teaching and learning arc welding tasks. It involves the use of a welding torch, an RGBD camera, an HDR camera, a VR headset, and image processing algorithms. By using this multi-sensor interface, the seam can be automatically located with 3D vision. The instant welding spot from the electric arc is recognized and immediate XR advice is provided to the user to improve the technique. Task scores, errors, and paths are displayed on the interface to provide the user with valuable feedback information. The effectiveness of the proposed method is experimentally validated with a group of beginners. Compared to the current practice, our method allows the instructor to have a clearer and more convenient way to demonstrate the process and quantify the trainees' performance. With the multi-sensor interface, users can independently practice the skill with minimal human supervision, hence, the proposed method has the potential to increase the number of trainees participating in a single session.

There are various limitations of the developed system. For example, only thick materials can be considered at present. A thin layer such as sheet metal is too small to be recognized, as the depth difference in the groove is insignificant for the algorithm to locate the seam and it may be misjudged as one large workpiece,

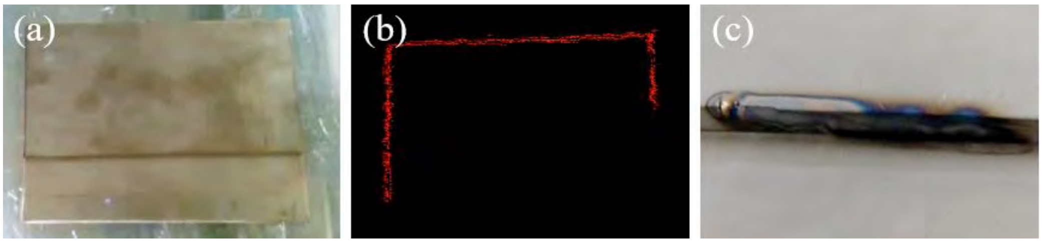


Figure 3.23: Some faulty examples. (a)–(b) Seam between sheet metal workpieces cannot be located as the depth difference is small; (c) Due to the limited angle of view from the system, the welding torch is hard to control with the system.

see e.g. Fig. 3.23; The current system works for grooves of least 5-mm thickness. The interface may also pose difficulties in controlling the angle of the welding torch due to the limited camera view, see e.g., in Fig. 3.23(c). This may lead to failures in joining two workpieces without a filler.

Chapter 4

Capability Sharing: Heterogeneous Multi-robot Collaboration

4.1 Introduction

In addition to the use of argumented perception capability in assisting humans for skill acquisition, robots can also assist and collaborate with other robots by leveraging and sharing their individual capabilities. For example, sharing the thermal perception information to another robot. Typically, multi-robot collaboration is treated as a task allocation problem, with researchers primarily focused on developing efficient task allocation algorithms. However, they often overlook the potential benefits of utilizing the capability features of heterogeneous robot teams [44], [45]. This approach can lead to hardware redundancy or over-reliance on a single robot, which may not fully optimize the available resources.

To address this, the concept of capability sharing should be considered to maximize the utilization of resources. For instance, rather than leaving idle capabili-

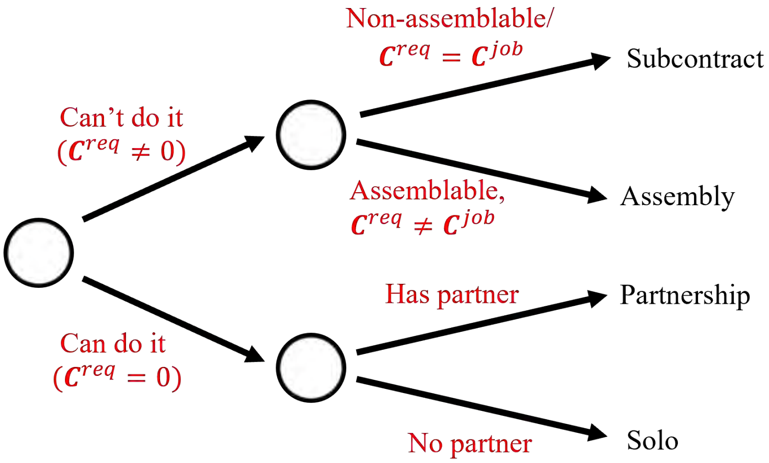


Figure 4.1: Decision tree for different working modes. If a robot can do the job alone, then the working mode is either solo or partnership. If a robot cannot do the job alone and is non-assemblable, it can only apply subcontract mode. If it is assemblable, the mode is assembly or subcontract.

ties unused or having a single robot take on the entire task [46], robots could share their available capabilities with one another. This chapter, therefore, focuses on enhancing task efficiency by fully utilizing the available capability.

Our aim in this chapter is to address these issues by developing a new distributed task allocation framework based on a semantic graph model. The proposed solution combines representation, enhancement and allocation methods of heterogeneous capabilities in robot teams. The original features of our new methodology are summarized as follows:

- a novel capability-based matrix masking method to represent heterogeneous robot configurations (thus, reducing hardware redundancy) and allocate their roles in a collaborative task;
- an effective method based on ontological dynamic knowledge graphs to model and locally optimize the motion and interaction between heteroge-

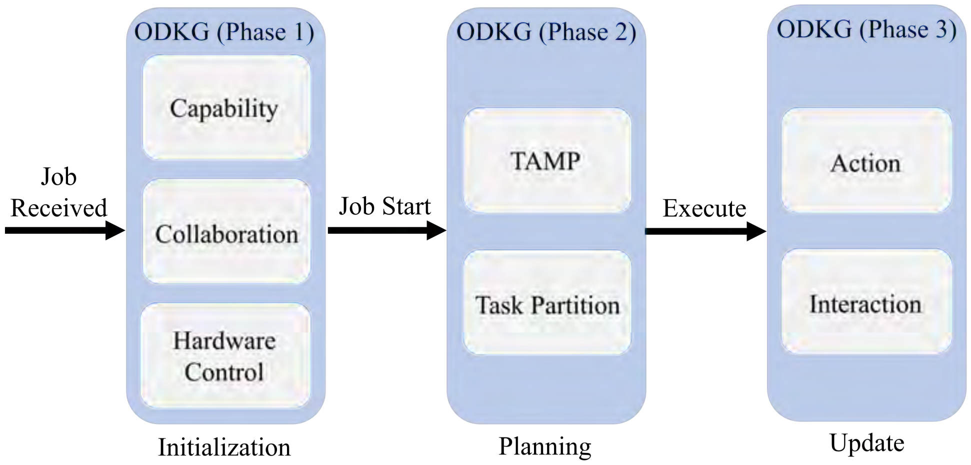


Figure 4.2: The full structure and use of the proposed ODKG. phase 1: initialization of fundamental setup for collaboration: phase 2: task planning for multi-robots; phase 3: information update.

neous robots;

- a graph adjacency approach that uses the least cost path to optimally utilize and coordinate resources in a group of robots.

The rest of this chapter is organized as follows: Sec. 4.3, 4.4, and 4.5 describe the methodology; Sec. 4.7 presents simulation and experimental results; Sec. 4.8 gives final discussions and conclusions.

4.2 Preliminaries

4.2.1 Notation

Matrices and column vectors are denoted by bold letters, such as $\mathbf{M}$ and $\mathbf{m}$. We use $[\mathbf{M}]_{ij}$ to denote the entry at the i-th row and j-th column of a matrix $\mathbf{M}$, and $[\mathbf{v}]_i$ to denote the i-th element of a vector $\mathbf{v}$. Throughout the chapter, the superscript $*^k$

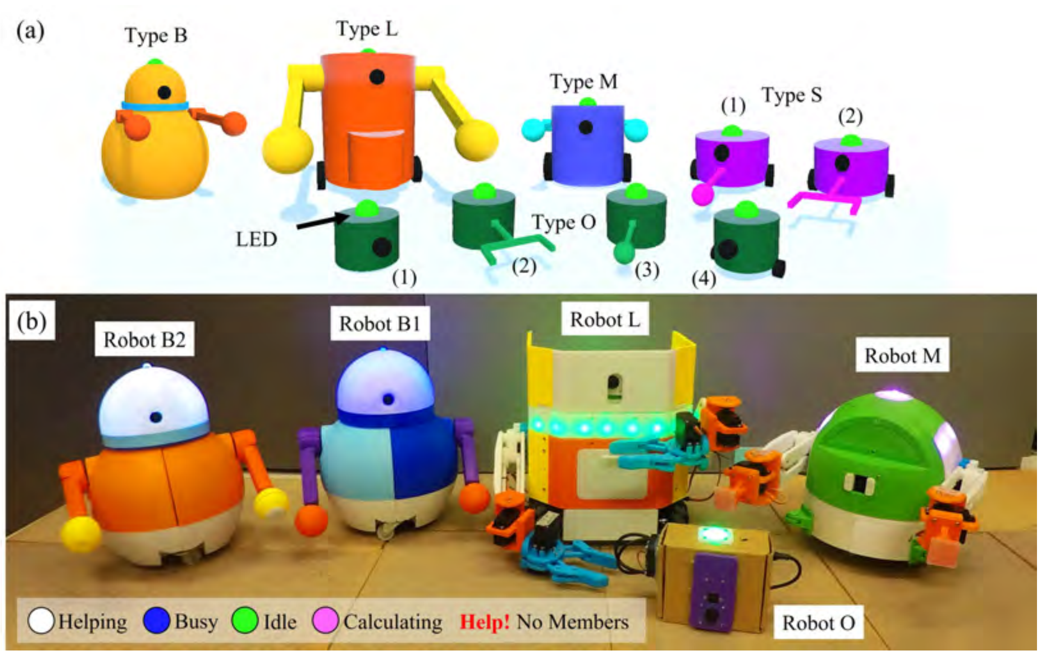


Figure 4.3: Heterogeneous robots developed for (a) simulation (b) experiments in our multi-robots society. Robot B1 and B2 are non-assemblable and others are assemblable. All of them have different talents and can work collaboratively. The LED on their heads indicates the current state. Green: idle state, no job or task. Blue: busy state, job received. White: helping state, granting favor as a teammate. In simulation: blue light with help: job received, no members and seeking favor.

is used to represent different instances of a model, e.g., $*^{self}$, $*^{job}$, $*^{req}$ represent the robot, job, and required versions of the structure $*$.

4.2.2 Definition of Terms

Roles: There are four roles in the proposed framework: 1) *Agent*: refers to the robot that receives the assigned job and which is responsible for completing it. 2) *Helper*: describes the robot that assists the *Agent* in completing the job. 3) *Team*: refers to various *Agents* and *Helpers* jointly working on the same job. A *Team* must contain one *Agent* and at least one *Helper*. 4) *Member*: Refers to any robots in a *Team*.

Capability: Refers to the configuration of a robot that is either pre-set in the factory or set after a hardware enhancement. In this chapter, we use the matrix **C** with various superscripts to represent the capability of a system/problem, e.g., $\mathbf{C}^k$ is used for the general case, $\mathbf{C}^{self}$ for a robot's own, and $\mathbf{C}^{job}$ for the capability demanded by a specific job.

In our method, we assume the distinctive hardware components are categorized into different capability domains, e.g., defined based on their nature. For that, we use the following three general capability domains: perception, manipulation, and locomotion. For example, standard 2D vision sensors and infrared cameras can both be grouped into a perceptual domain while grippers belong to the manipulation domain.

Types of Collaboration: In our method, we consider four working modes (see Fig. 4.1) to accomplish a mission, namely, *Solo*, *Partnership*, *Assembly* and *Subcontract*. The *Solo* and *Partnership* modes are for situations where the capability of a single robot is sufficient for carrying out the job. When there is only one robot around, the robot will act as an *Agent* and work in a *Solo* mode. When there are other robots available, they can act as *Helpers* and form a *Team* with the *Agent* to collaborate and share the workload under the *Partnership* mode. The difference between *Solo* and *Partnership* modes is the number of robots involved. If there is only a single robot, then, it belongs to *Solo* mode. Otherwise, it is a *Partnership* mode and the workload is split equally.

The capability of a single robot may not be able to match the demands of the job. Therefore, the *Assembly* mode allows robots to create larger structures with enhanced capabilities that fulfil the requirements. In this mode, the robot receiving

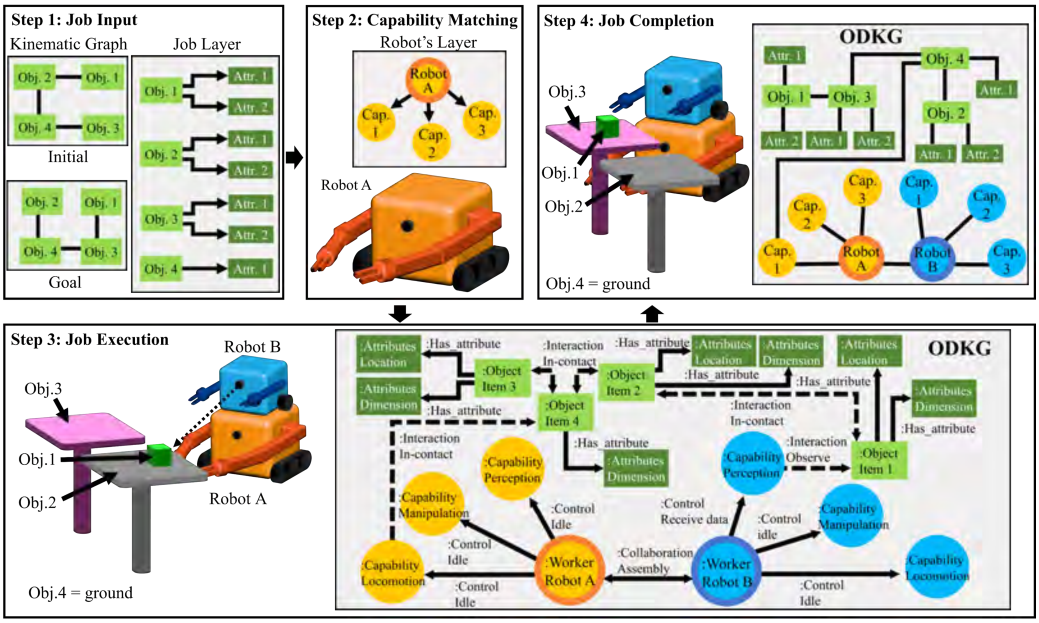


Figure 4.4: Example of the implementation of ODKG in the MRS with a pick-and-place task. At the beginning (Step 1: Job Input), kinematic graphs and job layer graph are sent to the agent when assigning a job. Then, the agent performs capability verification in step 2 and collaborates with the optimal neighbor in Step 3. An ODKG is constructed for task execution and is composed of a Robot Layer and a Job Layer. A full ODKG is illustrated in step 3. During the task execution in steps 3-4, the ODKG evolves with the latest interaction information between robots and the environments. When the job is completed (e.g. an object is moved to the targeted table as illustrated in step 4), a connected ODKG is formed and the kinematic relationships between objects are identical to the goal stated in step 1. A simplified version of a connected ODKG is shown in 'Step 4: Job Completion'.

the job needs to be assemblable. For a non-assemblable robot, it can instead adopt the *Subcontract* mode and transfer the job to other robots. Neighbours who agree to help will then act in *Solo* mode. For example, if a robot has more than 30% of battery power and has no current job/task assignment, it is defined as an *idle robot*. If a helper is running out of battery or the agent loses communication with its helper for a period of time, then, the agent losses a helper and may require to find another robot to continue the task.

4.3 Ontological Dynamic Knowledge Graph (ODKG)

4.3.1 Graph Theory

Graphs are structures that model relations between different objects. We denote a graph with the following pair $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, for $\mathbb{V} = \{v_1, \dots, v_n\}$ as a set of vertices, $\mathbb{E} = \{v_i v_j | v_i, v_j \in \mathbb{V}; i, j = 1, \dots, n\}$ as a set of edges, v_i as the i-th vertex, and n as the number of vertices. The neighborhood of v_i is defined as $\mathbb{N}_i = \{v_j \in \mathbb{V} | v_i v_j \in \mathbb{E}\}$ [66]. For an undirected graph $\mathbb{G}$, its degree matrix is defined as $\mathbf{D}(\mathbb{G}) = \mathrm{diag}[d(v_1), \dots, d(v_n)]$, where $d(v_i)$ is the degree of the vertex v_i and equal to the cardinality of the neighborhood $\mathbb{N}_i$. The graph's adjacency matrix $\mathbf{A}(\mathbb{G}) \in \mathbb{R}^{n \times n}$ satisfies:

$$[\mathbf{A}]_{ij} = \begin{cases} 1 & \text{if } v_i v_j \in \mathbb{E} \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

With the degree matrix and the adjacency matrix, we compute the Laplacian matrix of a graph $\mathbb{G}$ as $\mathbf{L}(\mathbb{G}) = \mathbf{D} - \mathbf{A}$, which is symmetric and positive semi-definite.

4.3.2 Application of ODKG in MRS

MRS usually work in highly dynamic/uncertain conditions, where changes can be external (e.g., obstacles and temperature in the environment) or internal (e.g., intermittent communication among agents). To perform tasks in these situations, MRS may need to build and track all semantic relations between the variable elements. For that, we adopt a dynamic knowledge graph to construct this information [38], [67]. The purpose of this dynamic knowledge graph is to provide a framework that models the heterogeneity of capabilities and allocates resources efficiently in the various working modes.

An *ontology* is a collection of branches describing the nature of objects, their attributes, and the relation among them [68], [69]. Their structure can be modeled as a hierarchical graph. In our method, we propose to use an ontological dynamic knowledge graph (ODKG) to represent the semantic information of the MRS during collaborative object manipulation tasks, see Fig. 4.2. When the robot receives a job, it enters Phase 1, which requires matching the capability, locating suitable members, and generating a hardware control schema (i.e. Phase 1 in Fig. 4.2). Then, it starts planning the action sequence to complete the job as illustrated in Phase 2 of Fig. 4.2. The robot executes the plan and updates the ODKG with the latest perceived information as shown in Phase 3 in Fig. 4.2. This approach is based on the following assumptions: 1) all robots are autonomous and capable of coordination via minimal local communication; 2) all robots can act upon the latest environment changes to collaboratively reach the target.

The proposed ODKG is a two-layer hierarchical structure that ontologically stores the information of internal physical components and the external environment, and models the interaction between them [65], [69], see Fig. 4.4. All the elements in the ODKG can be updated, created, and deleted based on the present observation/interaction [69]. The ODKG is a digital twin of reality but expressed in a graphical-semantic model [64]. We model the problem with two graph layers representing the Robot Layer $\mathbb{G}^R$ and Job Layer $\mathbb{G}^{job}$, respectively. We use the case depicted in Fig. 4.4 to exemplify the structure of the proposed knowledge graph.

Robot Layer. This layer represents the state of the MRS, i.e., the collaboration status of the robots and their components. It consists of the robots and their

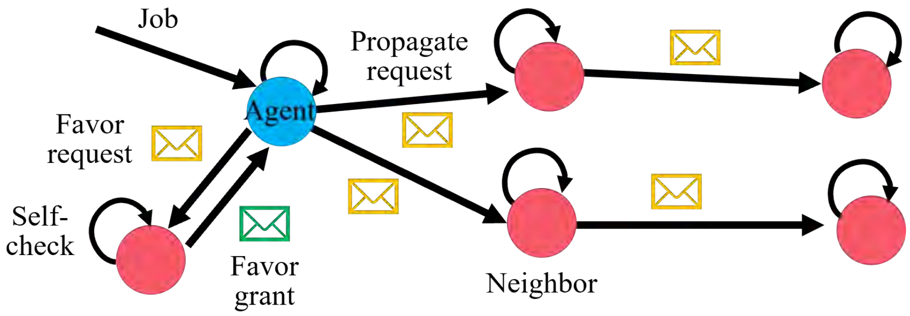


Figure 4.5: Multi-worker systems with a job assignment. Each node represents a worker. The blue one denotes the agent. The yellow envelope indicates the favor request message. The green envelope indicates the favor granting reply message. Once a job is passed to a worker, a self-capability check is conducted before spreading the request to neighbors. If the neighbor cannot do it and has not time-out yet, it propagates the request to its neighbors.

attributes, which are modelled as nodes in the graph $\mathbb{G}^R$. The attributes refer to the hardware components equipped in the robots. The collection of attributes defines the robot's specific capability. The relations between robots and between the robots and their attributes are represented as edges in the graph $\mathbb{G}^R$. We define the undirected edge between robots as a 'collaboration relation' and the directed edge between a robot and its attribute as a 'control relation'. An example of the Robot Layer is shown in Fig. 4.4.

Job Layer. The objects/items involved in a job as well as its attributes are included in this layer. Key information about the object (e.g., its dimension, weight, location) is encoded into the attributes. In the context of Task and Motion Planning (TAMP), this layer represents an extended kinematic graph that contains physical relationships between objects in the world and includes the necessary attributes of objects. The relation between objects can be changed by the robots (simply by interacting with them) and hence is defined as an 'interaction'. An attribute is linked to an object, and such relation is named 'has_attribute'. In Fig. 4.4, a simplified example of an individual job layer is shown in step 1 and a detailed job layer in an ODKG is shown in step 3.

Cross-Layer Relation. Connections between two items of different layers are defined as an 'interaction'. Physical interaction with an object is an 'in-contact' type of interaction while non-physical interaction is defined by its nature, such as 'observe' for perception.

Once a job is allocated to an agent, a kinematic graph of the current state $\mathbb{G}^{job}_{initial}$ and the final goal state $\mathbb{G}^{job}_{goal}$ with the required capability is passed to the agent, as illustrated in Fig. 4.5. Then, we combine $\mathbb{G}^{R}$ and $\mathbb{G}^{job}_{initial}$ to form the present state of the ODKG, see Fig. 4.4. Based on the job description, the agent then generates the job execution plan by TAMP [70] with task partition [71]. Members of the same team will undertake different tasks with the optimally selected components. The state transition planning to achieve the goal is formulated with TAMP in Sec. 4.6.2.

This graphical storage method facilitates the information update process between multiple workers [72]. As different layers have their own uniqueness, they can be viewed and utilized independently for the purpose of simplicity. For example, $\mathbb{G}^{R}$ can be used in the hardware control of robots, and $\mathbb{G}^{job}$ can be used to check the job progress. In this chapter, we assume all manipulable job objects are virtually connected in $\mathbb{G}^{job}$ when the job is completed.

The dynamic connectivity of the graph can reflect the current job progress based on the previous interaction history. When the job is in progress (i.e. a manipulable job object is being transported/manipulated), the job layer is a disconnected graph. Job completeness can be determined based on the connectivity in the Job Layer. The skeleton of the graph in each layer can be expressed with the

Laplacian matrix $\mathbf{L}(\mathbb{G}^{job})$ [66], [73]. $\mathbf{L}(\mathbb{G}^{job})$ is always positive semi-definite. Its smallest eigenvalue is always zero. Thus, connectivity is determined by evaluating the second smallest eigenvalue. If a connected graph is formed (i.e. the second smallest eigenvalue is larger than zero) and the kinematic relation between all the job objects is the same as the goal $\mathbb{G}^{job}_{goal}$, then, the job is completed.

We here introduce two methods to represent capabilities and match capabilities. Based on their nature, we refer to these methods as (1) Passive approach and (2) Active Distributed approach. Both methods are dependent on the robot layer in ODKG. The first one adopts a waiting (hence passive) strategy of the agents towards collaboration, while the second one uses a constant search (hence active) for the best collaboration strategy. Task planning and job execution processes of both methods are identical and only depend on the ODKG.

4.4 Passive Distributed Capability Matching

4.4.1 Strength-Based Capability Modelling — How strong am I?

To construct an ODKG for heterogeneous robots, we first model the configuration of each robot, which is equipped with (possibly) different hardware components. The capacity of each component is quantified by a *strength* metric whose magnitude encodes the capability to perform a given task. Large values signify a higher (or more specialized) capability.

In our proposed model, robots are either: (1) Assemblable, i.e., capable to form structures with other assemblable robots, or (2) non-assemblable, which can only work in parallel with other systems. Both non-assemblable and assemblable robots are capable to perform certain (typically simple) tasks, however, the strength level required to perform a challenging job could exceed that of a single robot. In this situation, a robot can collaborate with others by undergoing a capability enhancement or by simply passing the job to others [38]. To deal with this issue, we propose a capability-based dynamic task allocation algorithm that enables the shared execution of challenging tasks by a team of heterogeneous robots.

Note that each capability is denoted as an attribute node that is connected to the robot object in $\mathbb{G}^R$. We model the robot's capabilities as a matrix $\mathbf{C}^{self} \in \mathbb{R}^{N\times 5}$, where $N = D \times n$, for D as the number of the capability domains, and n as the number of strength levels. The proposed capability matrix is constructed with the following three terms:

$$\mathbf{C}^{self} = \left[\mathbf{s}^{self}\ \mathbf{q}^{self}\ \mathbf{P}^{self}\right] \tag{4.2}$$

where the strength vector $\mathbf{s}^{self} \in \mathbb{R}^{N\times 1}$ stores the intensity levels for the D capabilities in the structure:

$$\mathbf{s}^{self} = \Bigg[\underbrace{s_1,\dots,s_n}_{\text{capability 1}}\ \dots\ \underbrace{s_{N-n+1},\dots,s_N}_{\text{capability } D}\Bigg]^\intercal \tag{4.3}$$

for s_i as the ith strength level. Within the same capability domain, the strength levels are listed in ascending order, i.e., $s_1 < \cdots < s_n$ for capability 1 and so on. The structure of this strength vector will act as an indicator signal for protocol agreement between different robots.

The quantity vector $\mathbf{q}^{self} = [q_1,\dots,q_N]^\intercal \in \mathbb{R}^{N\times 1}$ contains the number of robot

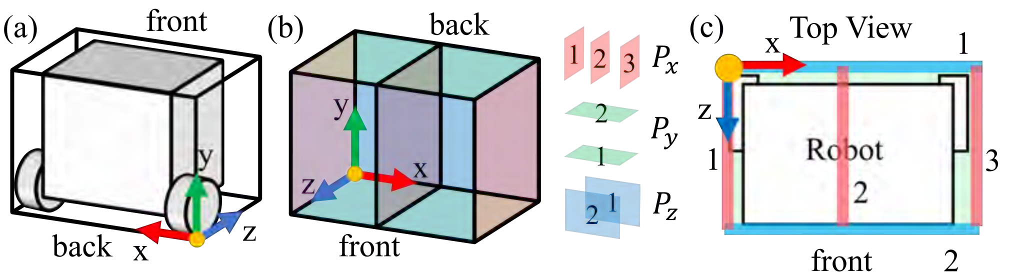


Figure 4.6: Coordination system of the component location. (a) Origin is denoted with the orange circle; (b) planes definition for each axis; starting from Plane 1; (c) top view of the robot with the coordination system with Plane 2 on the y-axis excluded.

components that belong to a specific capability domain, which are here denoted by d. In our model, components with a high strength level are capable of executing those tasks demanding a lower level within the same capability domain. Therefore, the quantity vector $\mathbf{q}^{self}$ is computed based on the following rule: $q_i = \text{count}(\text{level}(comp.) \geq s_i)$, where q_i equals the number of components with a strength level that is no smaller than s_i. e.g., if a robot has two components in the first capability domain whose strength levels are s_1 and s_2, thus, the quantity is $q_1 = 2$ and $q_2 = 1$.

The position matrix $\mathbf{P}^{self} \in \mathbb{R}^{N \times 3}$ stores the locations of the robot components and has the following structure:

$$\mathbf{P}^{self} = \begin{bmatrix} \mathbf{p}_x^{self} & \mathbf{p}_y^{self} & \mathbf{p}_z^{self} \end{bmatrix} \tag{4.4}$$

where the N-dimensional column vectors $\mathbf{p}_x^{self}$, $\mathbf{p}_y^{self}$ and $\mathbf{p}_x^{self}$ contain the x-axis, y-axis and z-axis coordinates of the robot components. We can write $\mathbf{P}^{self}$ in the following form:

$$\mathbf{P}^{self} = \begin{bmatrix} \mathbf{r}_1^{self} & \mathbf{r}_2^{self} & \dots & \mathbf{r}_N^{self} \end{bmatrix}^{\mathsf{T}} \tag{4.5}$$

where the row vectors $\mathbf{r}_i^{self} = [x_i, y_i, z_i]^{\mathsf{T}}$ denote the 3D position of each component. In our model, we define the coordinate frame's origin such that the components' positions are always represented with positive values. We assume that these locations fall within a set of pre-specified integer values, as conceptually represented as planes in Fig. 4.6. The lowest plane in the proposed model is Plane 1, therefore, the location is always greater than or equal to 1 for any component, and 0 for non-existing components. If there is more than one component with the same strength level, only the most distant from Plane 1 is considered in the position structure.

4.4.2 Capability Verification — Can I do the Job?

The 'Job' is assigned randomly to the agent. As shown in Fig. 4.1, once a job is passed to an agent, capability verification is carried out to determine the operation mode. A job is represented with a similar structure to (4.2), instantiated as $\mathbf{C}^{job} = [\mathbf{s}^{job}, \mathbf{q}^{job}, \mathbf{P}^{job}]$. We can verify the suitability of the agent to perform the task by comparing the quantity of the components $\mathbf{q}^{job}$ required by the job with $\mathbf{q}^{self}$. This is done with a simple Rectified Linear Unit (ReLU) [74] as:

$$[\Delta\mathbf{q}]_i = \max(([\mathbf{q}]_i^{job} - [\mathbf{q}]_i^{self}), 0) \tag{4.6}$$

The elements in the vector $\Delta\mathbf{q}$ describe the missing components that the agent needs to find to perform the job.

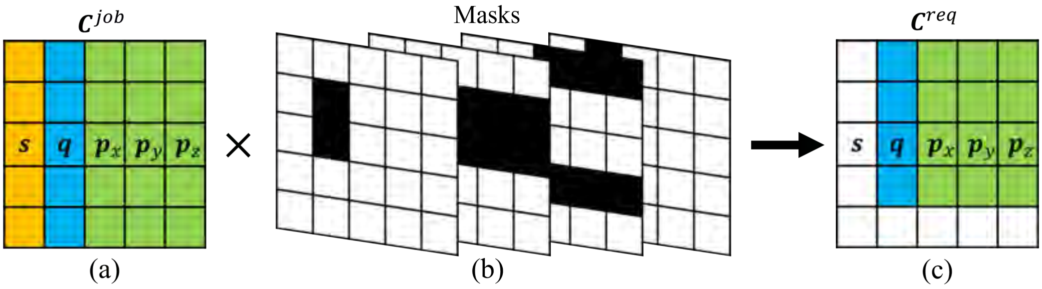


Figure 4.7: Capability verification visualization. (a) Capability of a job $\mathbf{C}^{job}$; (b) position and Quantity Masks; (c) results $\mathbf{C}^{req}$ after applying masks on (a); only colored areas carry meaningful information; white refers to redundant data, i.e. 0.

The location verification between $\mathbf{P}^{job}$ and $\mathbf{P}^{self}$ of components in each capability domain if similarly performed with ReLu. The required components location $\mathbf{P}^{req}$ is computed as:

$$[\mathbf{r}^{req}_{k,j}]_i = \max(([\mathbf{r}^{job}_{k,j}]_i - [\mathbf{r}^{self}_{k,min}]_i), 0) \tag{4.7}$$

$$\mathbf{P}^{req} = [\mathbf{r}^{req}_1 \; \dots \; \mathbf{r}^{req}_N]^\intercal = [\mathbf{p}^{req}_x \; \mathbf{p}^{req}_y \; \mathbf{p}^{req}_z] \tag{4.8}$$

where $\mathbf{r}^{job}_{k,j}$ is the jth row vector of the kth capability domain. $\mathbf{r}^{self}_{k,min}$ is the vector of $\mathbf{P}^{self}$ that is the closest to $\mathbf{r}^{job}_{k,j}$, hence, best meets the specifications of the job. The required components location for a $Helper$ is specified in $\mathbf{P}^{req}$.

To simplify calculations, our model adopts a binary masking approach [75], where masks are constructed to cross-map the robots' characteristics with the demanded job capability $\mathbf{C}^{job}$, see Fig. 4.7 for a conceptual representation. To determine the quantity of the location of each required component, we need to link the demand in $\mathbf{P}^{req}$ with $\mathbf{q}^{job}$ by constructing a binary mask of $\mathbf{P}^{req}$. To this end, we convert $\mathbf{P}^{req}$ into a vector with the same dimension as $\mathbf{q}^{job}$, and denote it as $\mathbf{m}^p$:

$$\mathbf{m}^p = \mathbf{P}^{req}\,\mathbf{e}_3 \in \mathbb{R}^{N\times 1} \tag{4.9}$$

where $\mathbf{e}_3 = [1,1,1]^\intercal$. To determine the number of final missing components (i.e. $\mathbf{q}^{req}$), we have to integrate the missing quantity $\Delta\mathbf{q}$ with the demanded quantity in terms of location $\mathbf{m}^p$. As the demand in position has a higher priority than that of the quantity needs, to avoid creating conflicting quantity results in $\mathbf{q}^{req}$, we cross-map the difference vector $\Delta\mathbf{q}$ with the mask $\mathbf{m}^p$. We use ${}^{\mathbb{B}}\{\mathbf{m}^p\}$ to normalize all terms in $\mathbf{m}^p$ greater than 0 to be 1, where ${}^{\mathbb{B}}\{*\}$ denotes the element-wise binarization operator. We combine ${}^{\mathbb{B}}\{\Delta\mathbf{q}\}$ with $\neg{}^{\mathbb{B}}\{\mathbf{m}^p\}$ by performing the logical 'and' operation. To this end, let us introduce the following term:

$$[\mathbf{b}]_i = [\Delta\mathbf{q}]_i([{}^{\mathbb{B}}\{\Delta\mathbf{q}\}]_i \wedge (\neg[{}^{\mathbb{B}}\{\mathbf{m}^p\}]_i)) \tag{4.10}$$

where $\neg$ and $\wedge$ denote the *not* and *and* logical operator, respectively. $\mathbf{b}$ is a quantity column vector in which the components represented in each $[\mathbf{b}]_i$ are only required from a quantity perspective, not a location perspective. To combine the quantity demands from $\mathbf{P}^{req}$ and $\mathbf{b}$, we apply the mask ${}^{\mathbb{B}}\{\mathbf{m}^p\}$ on $\mathbf{q}^{job}$ and combine the results with $\mathbf{b}$ to solve $\mathbf{q}^{req}$ as follows:

$$[\mathbf{q}^{req}]_i = [\mathbf{q}^{job}]_i[{}^{\mathbb{B}}\{\mathbf{m}^p\}]_i + [\mathbf{b}]_i \tag{4.11}$$

where $\mathbf{q}^{req}$ is a vector that integrates the missing components' quantity in terms of both location and quantity. Following (4.2), the final required capability $\mathbf{C}^{req}$ is constructed as:

$$\mathbf{C}^{req} = \begin{bmatrix} \mathbf{s}^{job} & \mathbf{q}^{req} & \mathbf{P}^{req} \end{bmatrix} \in \mathbb{R}^{N\times 5} \tag{4.12}$$

Once $\mathbf{C}^{req}$ has been determined by the agent, the working/collaboration format is determined following the rules described in Fig. 4.1. We now present a representative case study to illustrate the proposed methodology.

Example: Consider that the capability requirements of a job $\mathbf{C}^{job}$ and the agent's capability configuration $\mathbf{C}^{self}$ are:

$$\mathbf{C}^{job} = \begin{bmatrix} 1 & 1 & 1 & 4 & 1 \\ 2 & 0 & 0 & 0 & 0 \end{bmatrix}, \mathbf{C}^{self} = \begin{bmatrix} 1 & 1 & 2 & 3 & 2 \\ 2 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Based on the structure (4.2), we use (4.6) to obtain $\Delta\mathbf{q}$ and (4.7)–(4.8) to compute $\mathbf{P}^{req}$ as follows:

$$\Delta\mathbf{q}^{\intercal} = \begin{bmatrix} \max(1-1,0) & \max(0-0,0) \end{bmatrix} = \begin{bmatrix} 0 & 0 \end{bmatrix} \tag{4.13}$$

$$\mathbf{P}^{req} = \begin{bmatrix} \max(1-2,0), \max(4-3,0), \max(1-2,0) \\ \max(0-0,0), \max(0-0,0), \max(0-0,0) \end{bmatrix} = \begin{bmatrix} 0,1,0 \\ 0,0,0 \end{bmatrix} \tag{4.14}$$

We calculate $\mathbf{m}^{p}$ with (4.14) by using (4.9) as follows: $\mathbf{m}^{p} = \mathbf{P}^{req}\mathbf{e}_3 = [1,0]^{\intercal}$, which is used to obtain the binary mask $^{\mathbb{B}}\{\mathbf{m}^{p}\} = [1,0]^{\intercal}$. The term $\mathbf{b}$ is computed by using (4.10) as follows $\mathbf{b} = [0(0 \wedge 0), 0(0 \wedge 1)]^{\intercal} = [0,0]^{\intercal}$. The required quantity vector can be calculated via (4.11) with the binary operations $\mathbf{q}^{req} = [1(1)+0, 0(0)+0]^{\intercal} = [1,0]^{\intercal}$. By combining $\mathbf{s}^{job} = [1\ 2]^{\intercal}$, $\mathbf{q}^{req}$ and $\mathbf{P}^{req}$ from (4.14), we can compute the final required capability matrix as shown below:

$$\mathbf{C}^{req} = \begin{bmatrix} \mathbf{s}^{job} & \mathbf{q}^{req} & \mathbf{P}^{req} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 2 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.15}$$

where $\mathbf{C}^{req}$ shows that the agent is looking for a helper with one component

($[\mathbf{q}^{req}]_1 = 1$) containing the following features: a strength level of 1 ($[\mathbf{s}^{job}]_1 = 1$), and a height of 1 unit ($[\mathbf{p}_y^{req}]_1 = 1$).

4.4.3 Favor Granting — Let me Give you a Hand

Once $\mathbf{C}^{req}$ is ready, the agent starts searching for members. The communication among the MRS is represented using standard computer network models [76], [77]. If a solo/partnership/subcontract working format is possible (see Fig. 4.1), the agent spreads $\mathbf{C}^{job}$ to its idle neighbors. If assembly mode is possible, the agent spreads $\mathbf{C}^{req}$ with $\mathbf{C}^{job}$.

If an idle robot receives the request, it evaluates the agreement protocol based on the structure of its strength vector $\mathbf{s}^{self}$, and if it is asynchronous, the robot carries out synchronization mapping with the incoming message. Afterwards, the robot computes the capability difference $\Delta\mathbf{C} = [\mathbf{s}^{job}, \Delta\mathbf{q}, \Delta\mathbf{P}]$ by applying (4.6)–(4.8) where $\Delta\mathbf{P} = \mathbf{P}^{req}$ in (4.8).

If $\Delta\mathbf{C} \neq [\mathbf{s}^{job}, \mathbf{0}_{N\times 4}]$, the neighbor lacks certain requested capabilities and forwards the incoming message to its neighbors. If $\Delta\mathbf{C} = [\mathbf{s}^{job}, \mathbf{0}_{N\times 4}]$, the robot replies with a favor granting message and sends back its own capability, which we denote as $\mathbf{C}^{fav}$, and the robot becomes a responsive neighbor.

4.4.4 Member Selection — Who will be the Best Helper?

The optimal member can be determined by comparing the capabilities of each responsive neighbor $\mathbf{C}^{fav}$ [78]. To compare matrices efficiently, we do not simply take the large matrix $\mathbf{C}^{fav}$ for calculation. Instead, we condense it to a column vector **k**:

$$[\mathbf{k}^k]_j = [\mathbf{s}^k]_j[\mathbf{q}^k]_j[\mathbf{p}_x^k + \mathbf{p}_y^k + \mathbf{p}_z^k]_j, \quad \text{for } k = job, req, fav \tag{4.16}$$

where we multiple the elements in $\mathbf{s}^k$ and $\mathbf{q}^k$ to amplify their priorities in $\mathbf{k}^k$. We use addition in $\mathbf{P}$ to lower the impact of each $\mathbf{p}_x, \mathbf{p}_y, \mathbf{p}_z$ to the overall condensation results. The similarity between the received favors and the requirement can be reflected by the Euclidean distance. We represent the similarity with the Euclidean distance score θ^k [79]:

$$\theta^k = \frac{1}{1 + ||\mathbf{k}^{fav} - \mathbf{k}^k||} \times 1000, \quad \text{for } k = job, req \tag{4.17}$$

where a shorter distance implies higher similarity.

The solution is obtained by finding the neighbor with the highest θ^k. The neighbor with the most similar capability to the demanded one is denoted as ω_o, and determined by:

$$\omega_o = \arg\max_{\omega}(\theta_1^k \dots \theta_{2a}^k), \quad \text{for } k = job, req \tag{4.18}$$

where ω is an active neighbor and a is the number of active neighbor. A favor acceptance acknowledgment is sent back to ω_o indicating its role. A timeout approach is used to indicate favor rejection [77].

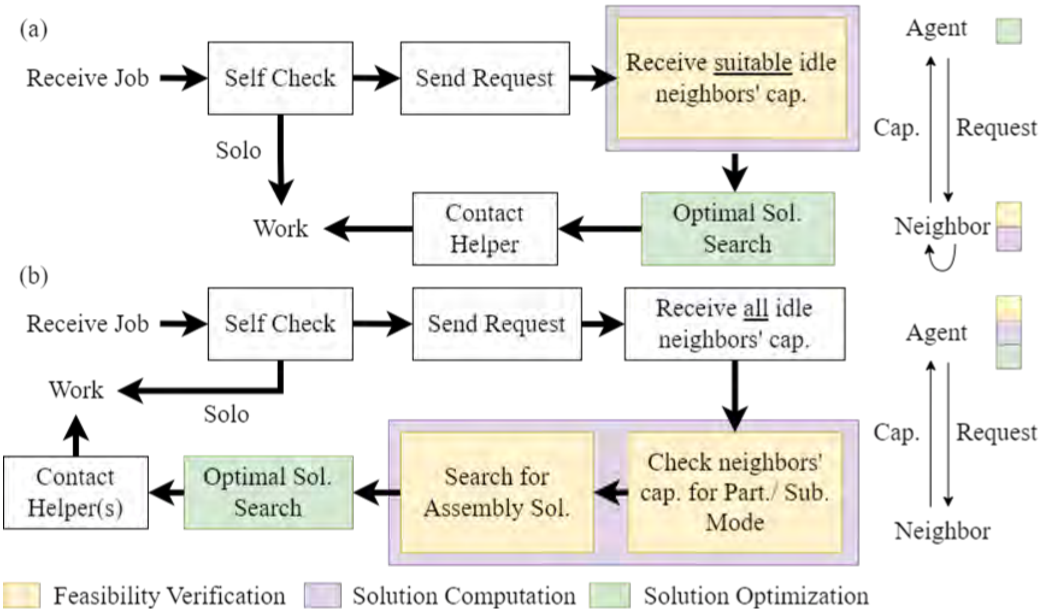


Figure 4.8: (a) Passive Distributed Method: agent only focuses on solution optimization with pre-processed solutions. (b) Active Distributed Method: agent actively takes part in all decision-making processes with all unprocessed neighbors' data.

4.5 Active Distributed Capability Matching

The method described in Section 4.4 uses a strength-based strategy to describe the *specialty* of a component (i.e. the differences between components in each capability domain, describing how strong is a given component) with a passive distributed approach, where neighbors take part in determining the operability of a task, and the agent is only responsible for finding the optimal solution among all available pre-calculated solutions that are passively received from neighbors. However, the passive method (which we shall refer to as Method 1) may not provide global optimal results that consider the combination of *all* idle neighbors. To obtain global optimal results, we propose an active distributed method.

4.5.1 Type-Based Capability Modelling

Components can also be categorized into different types. There might be a situation where the types of components are considered instead of the strength level, such that each type of component can only carry out one type of job (e.g. some thermal cameras provide both thermal and RGB images while others only provide thermal information). To this end, we introduce an active distributed type-based capability approach, which we shall refer to as Method 2. A conceptual comparison between these two methodologies is illustrated in Fig. 4.8. Method 1 can provide local optimal results with a shorter elapsed time in capability matching. Method 2 can provide global optimal results but the elapsed time is longer. Hence, the trade-off is time with optimality.

We define the type-based specialty and quantity in new formats and denote them as $\mathbf{T}^k = [\mathbf{t}_1^k, \cdots, \mathbf{t}_D^k]^\intercal$ and $\hat{\mathbf{q}}^k = [\hat{q}_1^k, \cdots, \hat{q}_D^k]^\intercal$ where D is the number of capability domains. Each row refers to one capability domain d, for $d \leq D$. For example, if there are type 1 and type 2 components in the capability domain d, then $\mathbf{t}_d^k = [1, 2]^\intercal$ and the corresponding quantity is $\hat{q}_d^k = [2]$. As the dimensions of different $\mathbf{t}_i^k$ may not match with each other, we append 0s to the end of $\mathbf{t}_i^k$ to make sure $\mathbf{T}^k$ is a matrix. The position $\hat{\mathbf{P}}_i^k$ of each component $\mathbf{t}_\mathbf{i}^\mathbf{k}$ is individually denoted as $\hat{\mathbf{P}}_i^k = [\hat{\mathbf{p}}_{i,x}^k\ \hat{\mathbf{p}}_{i,y}^k\ \hat{\mathbf{p}}_{i,z}^k]$. The position of all components is represented as $[[\hat{\mathbf{P}}_1^k]^\intercal\ [\hat{\mathbf{P}}_2^k]^\intercal \ldots]^\intercal$.

4.5.2 Capability Verification

The agent runs capability verification when it receives a job. Elements in $\mathbf{T}^{job}$ are compared with $\mathbf{T}^{self}$ row-by-row. If an element from $\mathbf{T}^{job}$ can also be found

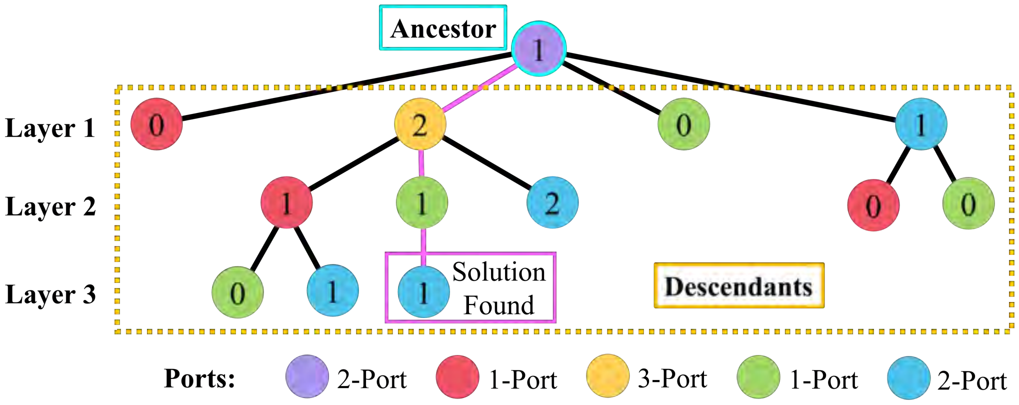


Figure 4.9: A modified assemble search tree. The purple node indicates the agent which is also the ancestor Υ. Others are descendants υ. Ports refer to the number of assembled ports in a robot. The number in each node shows the currently available node remains. The search ends when a solution is found in layer 3.

in $\mathbf{T}^{self}$, it is considered as specialty matched. If there is an element in $\mathbf{T}^{self}$ that fails to meet the requirement, the agent cannot work alone. Similar to Sec. 4.4.2, ReLu is used for quantity comparison. The quantity difference (i.e. $\Delta\hat{\mathbf{q}}$) is computed as in 4.6. If all the values in the result $\Delta\hat{\mathbf{q}}$ are equal to zero, solo working is feasible in terms of the quantity, otherwise, a co-worker is required. The agent sends a capability request to its neighbors but does not disclose any details of the assignment. Idle neighbors reply with their configurations $\mathbf{T}^{\omega}$, $\hat{\mathbf{q}}^{\omega}$, and the corresponding $\hat{\mathbf{P}}^{\omega}$ without any pre-calculation where we denote idle and replied neighbors as active neighbors (i.e. ω).

4.5.3 Collaboration Determination

In the following, ‘solution’ can be referred as possible collaboration results. It can be a single robot for partnership/subcontract mode or a list of robots for assembly. To calculate the optimal solution, the agent makes all the decisions actively based on all the received neighbors' capabilities. The position of a component is compared with ReLu when quantity and type are matched. If the result obtained from ReLu is not equal to 0, the component does not fit the requirement. If it is equal to 0, the component satisfies the requirement. All the neighbor's capabilities undergo a partnership and subcontract collaboration checking with the same capability verification stated in Sec. 4.5.2. If there is any neighbor who fits the job criteria, it implies partnership/subcontract collaboration is available and it is one of the solutions. For capability enhancement, having more neighbors means a larger amount of combinations have to be analyzed, which is computationally costly and time-consuming. Inspired by the assembly approach used in [40], we modify the breadth-first search algorithm in tree topology to determine the optimal capability enhancement solution as depicted in Fig. 4.9. We refer this tree as the modified assemble search tree. The agent constructs this tree after it receives responses from its neighbors.

In this tree topology, we use a node to represent a robot. The agent is the ancestor Υ while the idle neighbors are descendants υ. Nodes involved in a path from Υ to any υ are the member of a team. Our goal is to obtain a path (i.e. a solution) for the optimal assemble combination that has no repeated or redundant robots in it.

Each robot has a limited number of assembly ports, here denoted as $\mathbb{P}$ and which is equal to the maximum number of degrees in its node. A υ in the tree is a node with at least one degree/assembly port, i.e. $\mathbb{P} > 0$. The number of available ports after assembly with the current node is the team's available ports $\mathbb{A}_{\mathbb{P}}$, which is calculated as:

$$\mathbb{A}_{\mathbb{P}} = \sum_{i=1}^{f} \mathbb{P}_i - 2(f-1) - g \tag{4.19}$$

where f is the number of nodes involved in the path from Υ to the current node. When two robots assemble, an assembly port from each robot will be used, hence, two ports are deducted from the remaining available ports (i.e., the second term in (4.19)). g represents the number of ground-contacted robots. For example, in a "tower-like" assembly (see Fig. 4.4) with robots attached on top of each other, a single robot is responsible for the team's mobility, thus, $g = 1$.

A descendant υ is created if and only if there is an assembly port left from its parent's layer and a new combination can be built. A solution check is carried out after a new layer is constructed. At the lowest level of υ, a path's capability is compared with the capability requirement if the path's $\mathbb{A}_{\mathbb{P}} \geq 0$. If there is a path satisfying the requirement, it is one of the solutions and the search is completed. A new layer will only be developed if there is no solution found in the current layer. An example of this modified search tree is shown in Fig. 4.9.

If more than one solution is available, we apply a capability similarity comparison to locate the optimal solution. Similar to (4.17), we define the Euclidean distance score $\hat{\theta}^i$ as follows:

$$\hat{\theta}^i = \frac{1}{1 + ||\hat{\mathbf{q}}^i - \hat{\mathbf{q}}^{job}||} \times 1000 \tag{4.20}$$

where we only focus on the number of components. The specialty is omitted here as a higher type index component does not imply it is better than that with a lower type index. i is an available solution that denotes the capability of an idle neighbor ω in partnership/subcontract mode or the overall capability of a robot

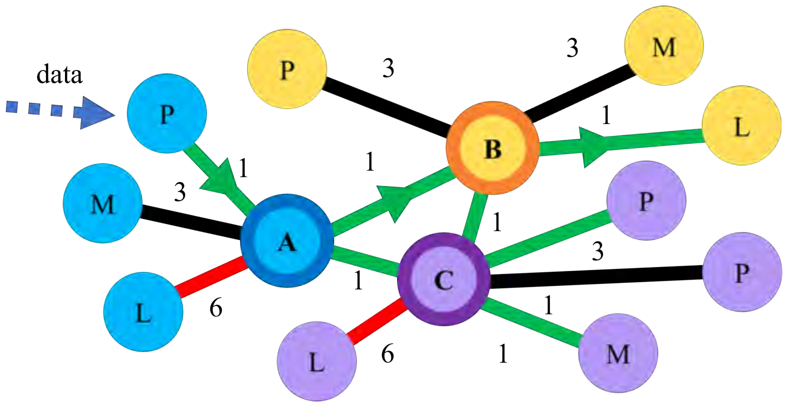


Figure 4.10: Example of hardware control when new sensory data arrives to a group of assembled robot team that contains A, B, and C workers. A locomotion capability for action is in request. ‘P’, ‘M’, and ‘L’ stands for perception, manipulation, and locomotion components. The edges in red, black, and green indicate the highest, medium, and lowest cost connections. The arrow indicates the flow of the action command resulting from the least cost path calculation.

team in assembly mode. The optimal member list can be reached with the largest $\hat{\theta}^i$ by (4.18).

4.6 Capability-Based Task Allocation

4.6.1 Control of a Team of Robots

To address the hardware coordination of a team of robots, we first construct a weighted graph $\mathbf{W}(\mathbb{G}^R)$ from our ODKG. $\mathbf{W}(\mathbb{G}^R)$ is an adjacency matrix but with the cost included for edges. We use cost to indicate the availability and priority of accessing a component. We set the range of cost to be $[1, 2m]$, where m is the total number of members in a team. If there is any component not available to be used freely in the collaboration assembly (e.g., locomotion of Robot B in Fig. 4.4), the highest cost (i.e. $2m$) is used in the edge between the robot node and the component node. If there is any component not requested by the agent/job, these components are treated as redundant with a cost equal to m in the linkage. Otherwise, the component is regarded as free and with the minimum cost (i.e. 1). This situation is visualized in Fig. 4.10.

When a robot detects new sensory information, it activates some functions/actions of the hardware components, such as moving or grasping. In this section, we focus on the hardware selection of each task, but not on task planning. Each capability has its own action set $\mathbb{A} = \{a_1, a_2, \dots\}$. To achieve accurate control, we find a minimum cost path with the Dijkstra algorithm [66], [80] from the current capability attribute node that receives the new sensory data to the capability attribute node that equips the required action a_i. Consider the situation shown in Fig. 4.10 as an example. When the perception component of Worker A receives a new useful observation, Worker A analyses the data and decides to take an action a_1, e.g., to move forward. This finds the minimum cost path from its 'P' node to a capability node that contains a_1 in its action set. From the node 'P' of Worker A to a locomotion capability node 'L' of Worker B, only 3 units of cost are involved, which is the lowest cost in the graph. Hence, the team controls the locomotion component of Worker B to execute the action a_1.

The proposed control mechanism can be used with all types of working modes. Its purpose is to ensure the action is taken by the correct component in a robot team without hardware conflicts. Fig. 4.10 conceptually depicts the approach.

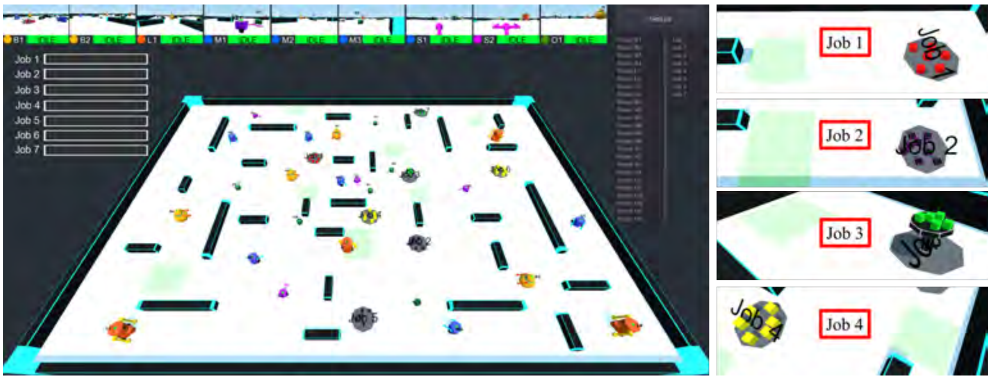


Figure 4.11: Simulated robot factory with an enlarged view of each job.

4.6.2 ODKG: Planning and Update

To determine the response for each sensory data and environment change, we plan the action with the ODKG. Coordinating task allocation with capability consideration is complicated. Therefore, we utilize Task and Motion Planning (TAMP) [70] with task partition [71] to generate a feasible action policy $\pi = \{a_1, a_2, \dots\}$ for low-level motion planning. We use the Force-Based Algorithm for Motion Planning [81] when implementing the distributed path planning on multi-agents for each action a_i. In the task planning stage, once a job is allocated to an agent, the current $\mathbb{G}^{job}_{initial}$ and the final goal environment state $\mathbb{G}^{job}_{goal}$ with the required capability is passed to the agent. We take the $\mathbb{G}^{job}_{initial}$ and $\mathbb{G}^{job}_{goal}$ as the input for TAMP framework.

The pose of all objects in the kinematic graph is with respect to a constant object in the world frame or to a random object. When there is a new interaction caused by an action a_i (e.g., ‘in-contact’ or ‘observe’), a new relationship between objects is created, and state transitions occur. A state, here denoted as $\mathtt{s}_i$, refers to the dynamic status of a group of variables. In our case, a state refers to the status of the environment and the robot-environment interaction. Each a_i drives the state

Table 4.1: Configuration of Robots (Method 1)

Types	Amount	Perc.	Man.	Loc.	Size	Coll.	Ports
B	2	1, lv 1	2, lv 1	lv 2	4	lv 1	—
L	1	1, lv 1	2, lv 1	lv 2	5	lv 4	3
M	3	1, lv 1	2, lv 1	lv 2	3	lv 3	2
S1	1	1, lv 1	1, lv 1	lv 1	2	lv 3	1
S2	1	1, lv 1	1, lv 2	lv 1	2	lv 3	1
O1	1	1, lv 3	—	—	2	lv 2	1
O2	1	—	1, lv 2	—	2	lv 2	1
O3	1	—	1, lv 1	—	2	lv 2	1
O4	1	1, lv 2	—	lv 1	2	lv 2	1

Table 4.2: Configuration of Robots (Method 2)

Types	Perc.	Man.	Loc.	Size	Coll.	Ports
B	1, type 1	2, type 1	type 1	4	type 1	—
L	1, type 1	2, type 1	type 1	5	type 1-4	3
M	1, type 1	2, type 1	type 1	3	type 1-3	2
S1	1, type 1	1, type 1	type 1	2	type 1-3	1
S2	1, type 1	1, type 2	type 1	2	type 1-3	1
O1	1, type 3	—	—	2	type 3	1
O2	—	1, type 2	—	2	type 2	1
O3	—	1, type 1	—	2	type 2	1
O4	1, type 2	—	type 1	2	type 2	1

$\mathtt{s}_i$ towards the next state $\mathtt{s}_{i+1}$ represented as $a = \langle \mathtt{s}_i, \mathtt{s}_{i+1} \rangle \in \Gamma$. After an action, the transition between the current state to the next state is denoted as $\Gamma \subseteq \mathtt{S} \times \mathtt{S}$, where Γ is the transitions, $\mathtt{S} = \{\mathtt{s}_0, \mathtt{s}_1, \dots\}$ is a set of states, $\mathtt{s}_0, \mathtt{s}_{goal} \subseteq \mathtt{S}$ are the initial and the goal state. We match the actions in π to the corresponding robot with $\mathbf{W}(\mathbb{G}^R)$ by task partition [71].

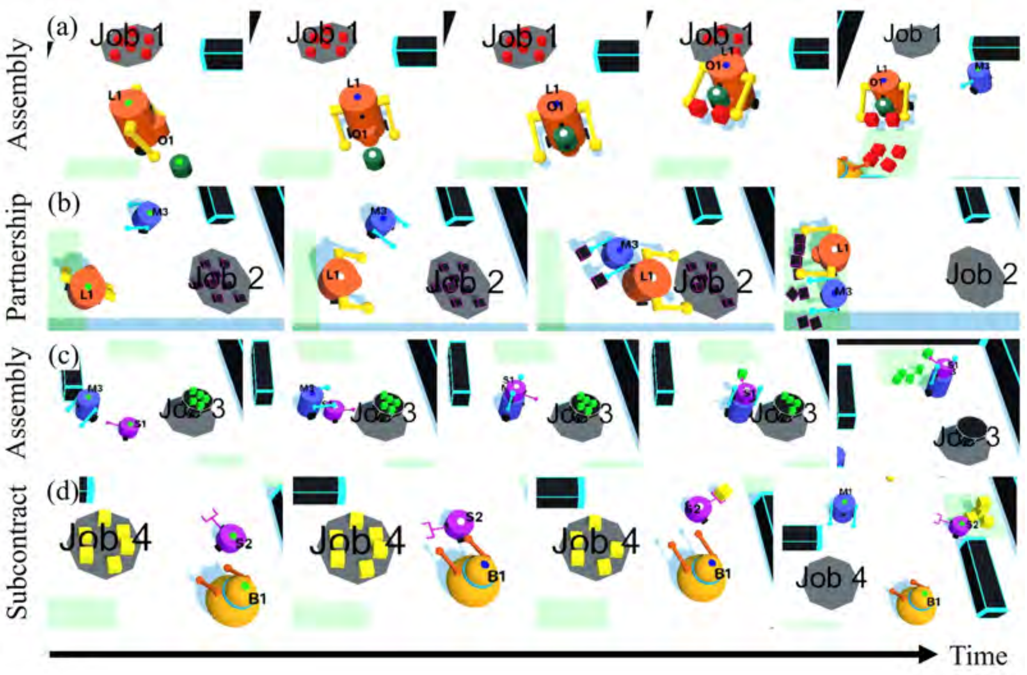


Figure 4.12: Demonstration of different types of collaboration working process: (a) Assembly mode in Job 1: the orange robot goes and picks up the green robot for assembly. (b) Partnership mode in Job 2: the orange robot helps the blue robot in partnership mode. (c) Assembly mode in Job 3(a): the blue robot assemblies with the purple robot to extend its height for reaching the green box. (d) Subcontract mode in Job 4: the yellow robot passes the job to the purple robot as only the purple robot can meet the job requirement.

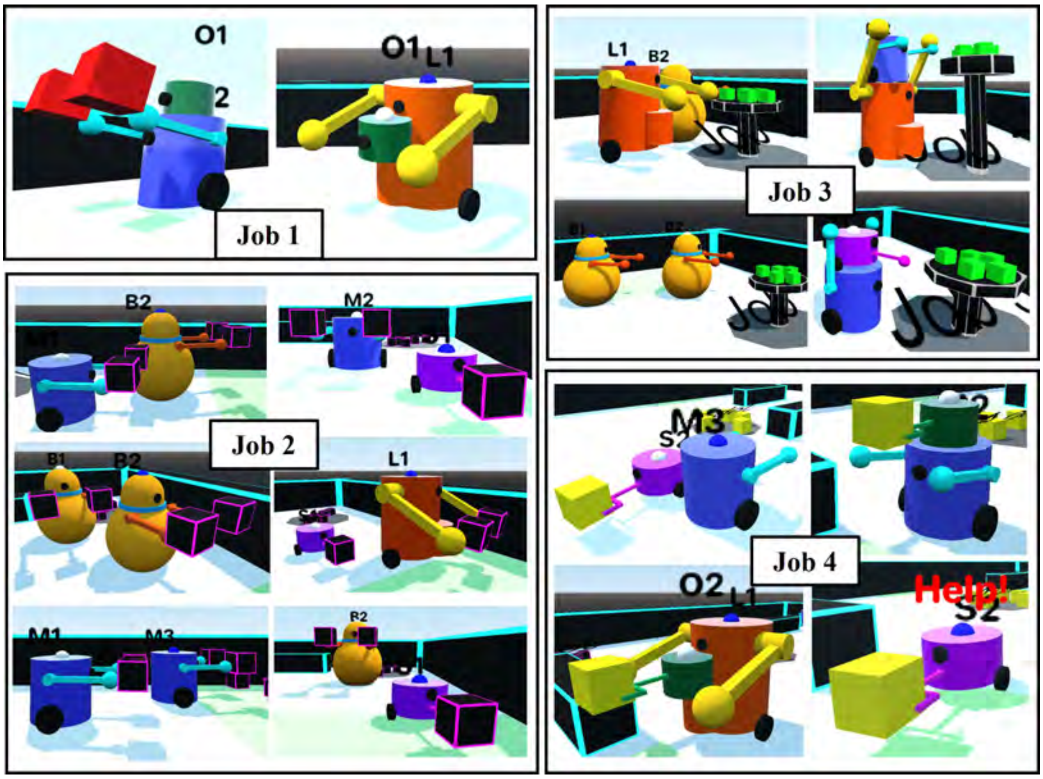


Figure 4.13: Example results of different combinations: Job 1: assembly; Job 2: partnership; Job 3: partnership and assembly; Job 4: solo and subcontract

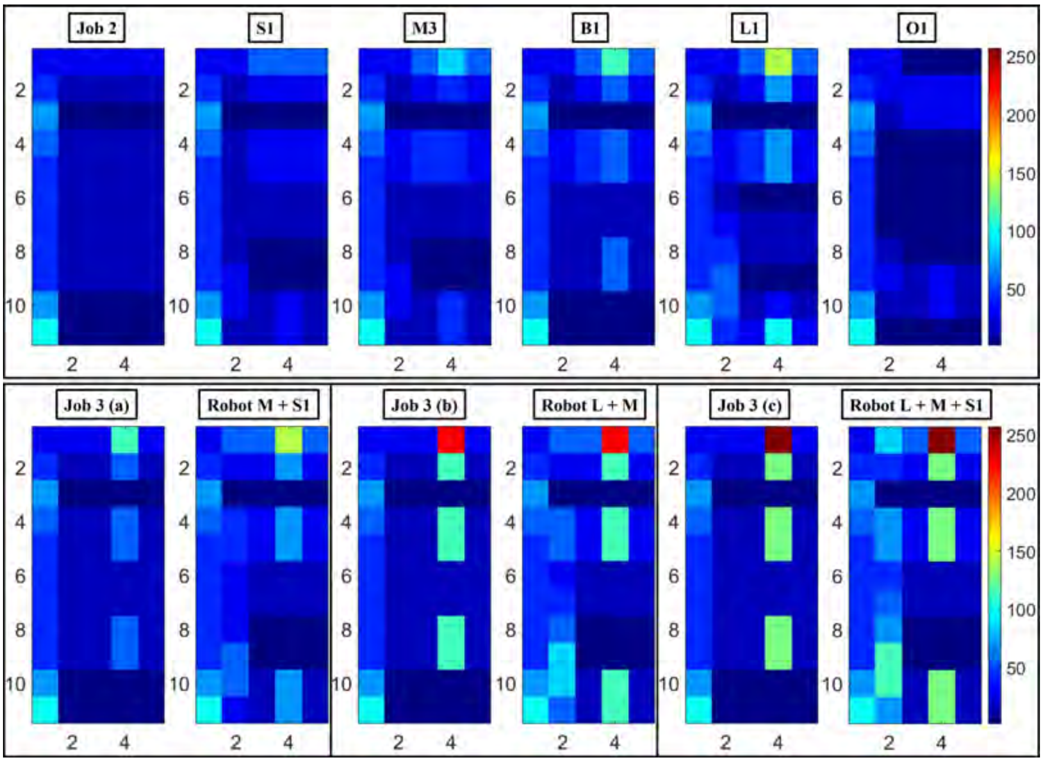


Figure 4.14: Strength-based capability **C** visualization for Job 2 and 3 with each robot and team are shown with the range of $[0,255]$. The larger number in the capability is, the higher the color index is. Job 3(a): 4-unit height; Job 3(b): 8-unit height; Job 3(c): 9-unit height.

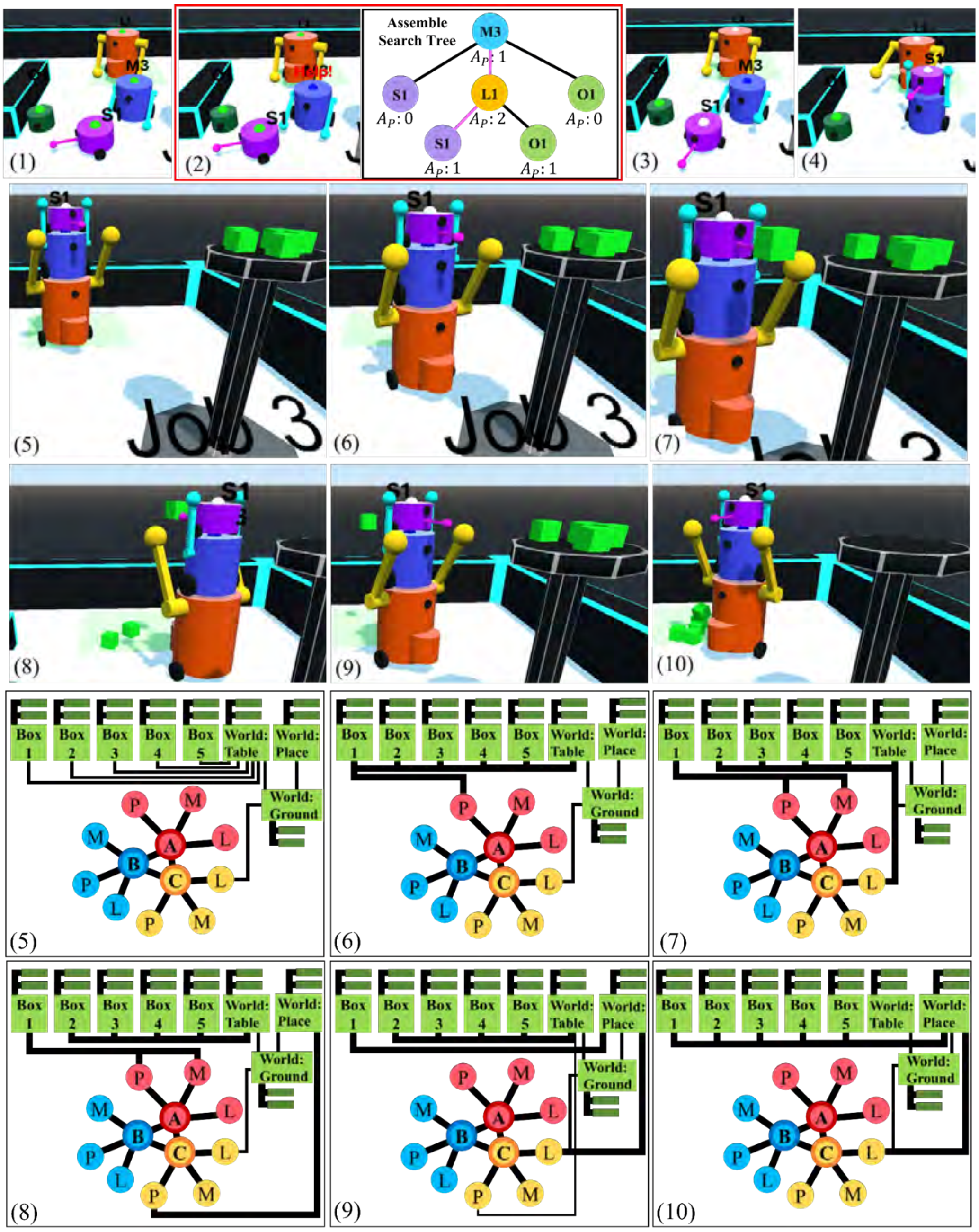


Figure 4.15: Multi-robot assembles. (1) Before receiving any job. (2) Job 3 is assigned to the blue one. Its LED turns blue with a red ”Help” word, indicating it is seeking help. A_P in the Assemble Search Tree refers to available port remains after the current assemble and the solution is indicated with pink lines. (3) The optimal solution is found. The LED on the purple and the orange robots turn white. (4) Assemble with reference to the solution. (5)-(10) Working and updating ODKG correspondingly where workers A, B, and C are the purple, blue, and orange robots.

4.7 Results

4.7.1 Simulation Setup

To validate the proposed methodology, we develop a small virtual robot world with a game engine (Godot). The system enables to simulate multiple robots that collaboratively perform different types of jobs (see Fig. 4.11). As this method is focused on decentralized multi-robot collaboration, all agents move randomly in the idle state and execute their own scripts independently. No centralized algorithm is used for coordination. Data is stored by the main system for our further analysis. We perform all simulations on a PC equipped with an RTX-3060 GPU.

A simple box pick-and-place task is chosen as the job. The mission is simple but allows testing the proposed method as each job has different capability demands. For instance, some jobs require a certain strength/type of specialty while some do not. A job can generally be carried out by various working modes. Different kinds of assembly combinations may also be available for the same job.

We design nine robot types with diverse capabilities (see Fig. 4.3). The amount and configuration of each robot type are indicated in Table 4.1 and 4.2, where “Amount”, “Perc.”, “Man.”, “Loc”, “Coll.”, and “Ports” refer to the number of robots in the field, perception devices, manipulation tools, locomotion units, collaborability, and assembly ports, respectively. In this numerical study, we focus on the ‘tower-like” assembly, where height is the most significant element in terms of dimension. “Size” refers to the height of a robot, and “lv” denotes level. Assemblability can be reflected in the number of available assembly ports that are shown in the last column. “1, lv 1” and “1, type 1” refers to one component with strength level 1 and one component with type 1 specialty respectively.

Four different job types are created. Job 1 focuses on the specialty of perception. It requires a level 3 vision device to detect heat and is designed to test the assembly collaboration where the assembly mode is the only available option. Job 2 focuses on the member selection of multi-robot collaboration, and it only requires level 1/type 1 component for each capability domain. Capability enhancement is unnecessary in this job type. Job 3 focuses on the robot's capability enhancement in height, where three different height constraints are examined: (a) 4-unit height, (b) 8-unit height, and (c) 9-unit height. Job 3 is designed to examine all kinds of collaboration formats mentioned in the previous section. There are two types of robots (Type B and Type L), rich in height that they can conduct Job 3(a) in solo/partnership mode. Thus, all working modes are potential solutions. Job 4 focuses on the specialty in the manipulation domain. Type S1 robot is the only robot that is capable of accomplishing Job 4 in solo mode, and the Type O2 robot can accomplish Job 4 but in assembly mode.

4.7.2 Capability Enhancement

Passive Distributed Methodology

Results of optimal member determination are presented in Fig. 4.12. In Fig. 4.12 and Fig. 4.13, robots are able to perform the optimal capability enhancement to satisfy the job requirement. Take Job 3(a) as an example, where there are only two robot types. B and L are capable to conduct the job in solo mode (see Fig. 4.14). Type M robot has to assemble with Type S1 robot to satisfy the requirement as shown in 4.12 (c). Through assembly collaboration, those incapable robots (i.e. Type M and Type S1 robots) can now complete Job 3 with help from others.

Active Distributed Methodology

To test the performance of the capability enhancement approach, we use Job 3(c) as an example where at least three robots' involvement is needed. Fig.4.14 shows that only when Robot L, M, and S1 assemble, their team capability of color level can reach the color level of the Job3(c) requirement. The assembly process is demonstrated in Fig. 4.15 (1)–(5). Four different robots are placed close to each other such that all of them are in the same neighborhood. When the blue Robot M3 receives a job, it calculates the feasibility of solo working. As Robot M3 cannot do it alone, it sends a help request to all idle neighbors which include orange Robot L1, purple Robot S1, and green Robot O1. To satisfy the job requirement, the agent computes the optimal solution which is to assemble with Robot L1 and S1. The assembly search tree results and the capability enhancement process is demonstrated in Fig. 4.15.

4.7.3 Collaboration Performance and Task Efficiency Analysis

Different member combinations are tested with our method, see Fig. 4.13. Robots can locate a suitable co-worker with an optimized capability to match results and form a team. Different working modes and collaboration strategies may lead to different task efficiency results. To investigate the differences, robots are randomly placed on the field and each combination is tested 10 times.

The task efficiency outcomes of various working modes are shown in Fig. 4.16. The shaded area shows the upper and lower boundary of the results. In Job 1, as there is only one type of feasible solution, if the agent is not an assemblable robot, the mission cannot be completed. In Job 2 and 3, it can be easily observed that partnership collaboration can reach the highest efficiency as more manpower can boost productivity. Solo mode is the second most efficient way to conduct the job as there is no extra computation involved in collaboration decision-making. Subcontract is the slowest workable solution in Job 2. Although subcontract is the sub-optimal way to work, it is still valid as the job is completed. The assembly mode in Job 3 is the most time-consuming as assembly takes time, such as approaching the agent and performing assembly. In Job 3 and 4, there is a large overlapping proportion between the assembly and the subcontract mode. The disparity between these two working modes is not significant. In Job 4, robots can either subcontract with a Type S2 robot or assemble with a Type F2 robot. Thus, the performance of solo, assembly and subcontract are similar.

It can be summarized that partnership is the best strategy in general for completing a project with the least amount of time spent. As there are no differences between the outcomes with the designed workable solutions, the model is valid in terms of the working mode.

4.7.4 Experimental Setup

To validate the feasibility of our proposed model, a small robot setup with 4 types of robots is developed (see Fig. 4.3 and 4.18). Robot B1 and B2 are Type B robots; Robot L is Type L; Robot M is Type M; Robot O and O1 are Type O1 robots; Robot O4 is a Type O4 robot. The color of the LEDs on the robot's body indicates their current status (see Fig. 4.3). In the experiment, robots first analyze their observed data and then pass the useful information to others. In this section, we focus on Job 1 and Job 3 with the assembly mode. The requirement

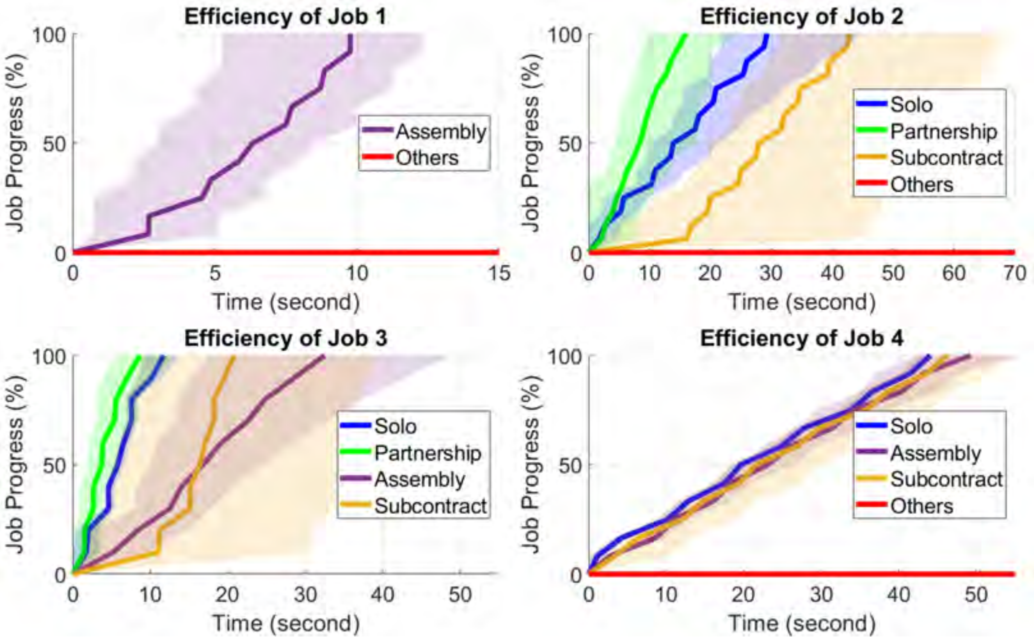


Figure 4.16: Task efficiency of Job 1-4 under diverse working modes.

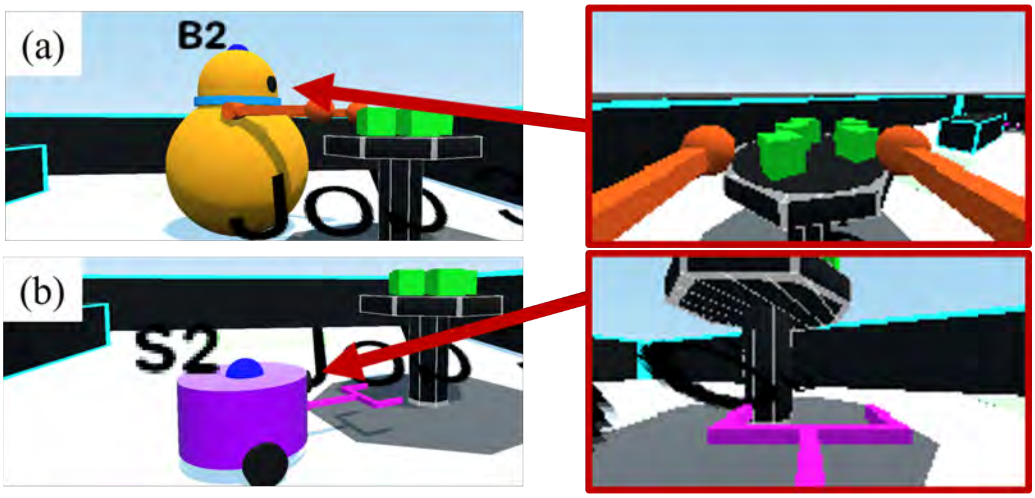


Figure 4.17: Comparison our method with the state-of-the-art method. (a) The agent with our method can reach the box on the table; (b) the agent with the state-of-the-art method cannot reach the box on the table.

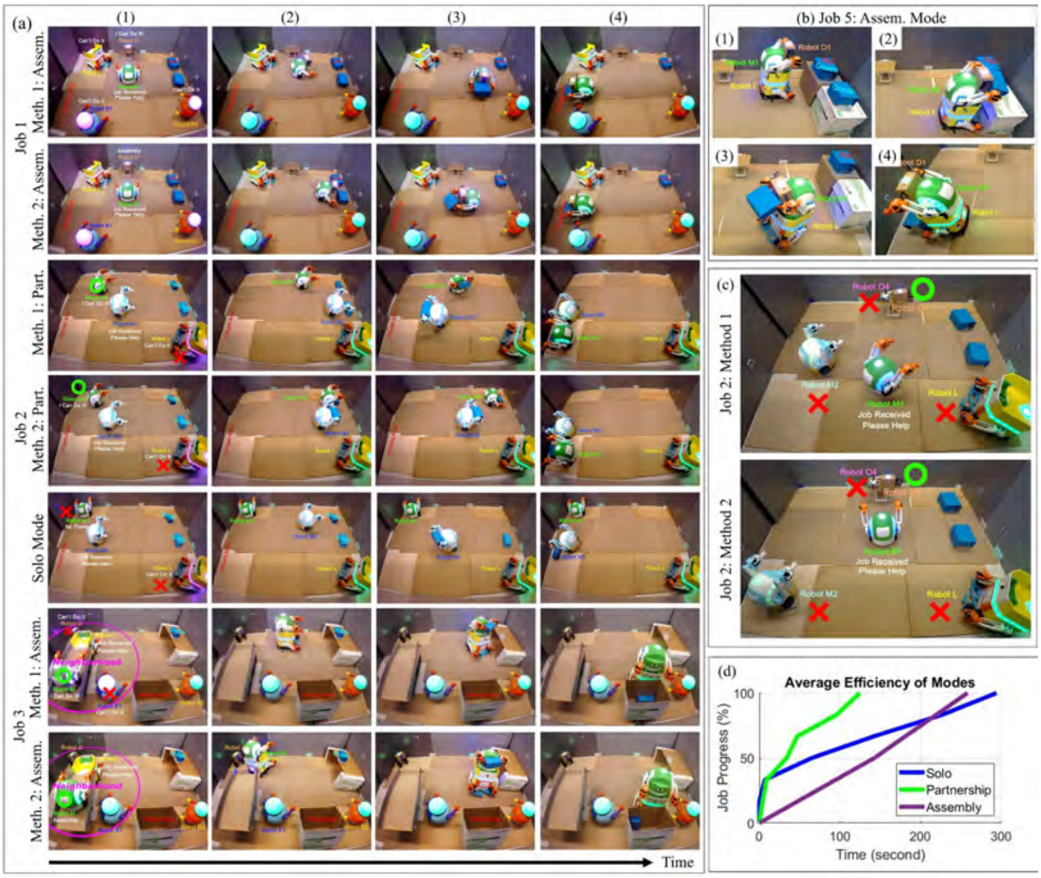


Figure 4.18: Experiment: Job 1-3 with different methods. (a)(1) The agent receives a job and seeks favor. (a)(2) – (5) start working with various working modes: solo, partnership, or assembly. (b) A new job 5 is included that which requires 8-unit of height and a level 3/type 3 perception capability to locate a non-hot box. (b)(1)–(4) Robot L assembly and work with Robot M and Robot O1. (c) Method 1 and 2 give the same member selection results when Job 2 is assigned to Robot M1. (d) The average task efficiency of different working modes.

of Job 1 and 3 are the same as the simulation. To make the experiment more relevant to the real-world application, Job 1 requires moving goods away from a 'simulated explosive' object. Job 3 requires picking goods from a table with a height of 8 units and placing them into a box. The robots use a Raspberry Pi 4b for processing and communication, and an Arduino for low-level control. The Job is assigned from an Ubuntu computer to the robots. All programs and decisions are executed individually.

4.7.5 Collaboration Analysis with Object Pick-and-Place Tasks

In both methods, the agents form the optimal strategy based on their capabilities. In Job 1, we hide a heater inside one of the blue boxes to simulate the existence of explosive material. All blue boxes are identical and cannot tell the difference from the outlook. The robot needs to acquire a heat detector to differentiate the safe box from the dangerous one. We assign the job to Robot M. In Fig. 4.18, Robot M transfers states from $\mathtt{s}_{initial}$ to $\mathtt{s}_{assemble}$ and to $\mathtt{s}_{goal}$. Robot M successfully conducts the assembly with Robot O. With the extended thermal vision from Robot O, the team is capable to select the correct box $\{a_{detect}, a_{grasp}\}$ and move it to the desired area $\{a_{move}, a_{release}\}$, away from the explosive box. The experiment proves the robot's ability to execute the job collaboratively and to utilize the optimal components via the proposed methodology.

In Job 3, the height constraint is unachievable for any single robot in the field. To test the optimal member localization and the assembly working mode between Robot L and M, we specifically assign the job to Robot L. In the experiment, Robot L assembles with the appropriate worker which is Robot M. Through the

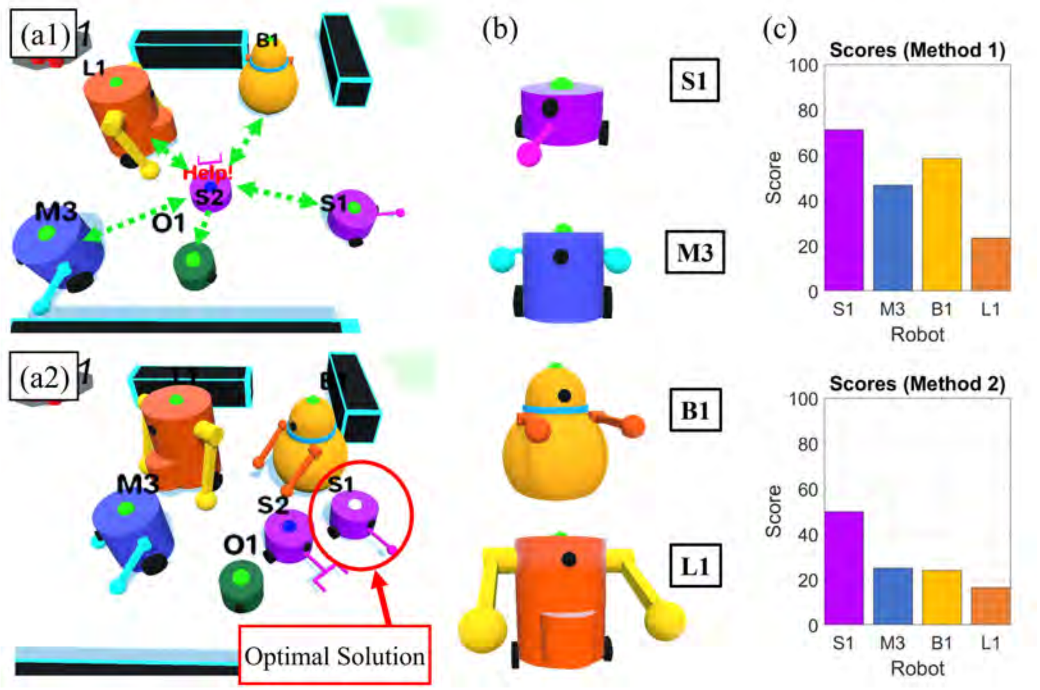


Figure 4.19: (a1) Robot S2 sends a favor request; (a2) robot S1 is selected in both methodologies. (b) Potential Members. (c) Similarity score θ for each potential member with Method 1 and 2.

implementation of the ODKG, the robot team can achieve optimal component control. The team uses the gripper from Robot M instead of that of Robot L, to grasp $\{a_{grasp}\}$ and unload $\{a_{release}\}$ the goods. With the capability enhancement, the team can pick up the goods from a table and put the goods into the target box for packing. Based on the results, it can be concluded that the proposed model is feasible and valid. It matches our observation in the simulation results. It also proves the potential for adopting the proposed model to solve the real-world multi-robot collaboration problem.

4.7.6 Performance Comparison

We numerically compare (i.e., with the simulation environment) the optimal performance of our proposed methods by placing multiple potential members around the agent. As shown in Fig. 4.19, all 6 types of robots are placed next to each other. We assign Job 2 to Robot S2. All the neighbors can conduct the job in solo mode, except Type O robots.

Passive Distributed Methodology

Job 2 requires a strength level 1 manipulation component. As Robot S2 is capable to carry out the job by itself, it looks for a partner to conduct the job in parallel and to boost productivity. The agent sends a favor request containing $\mathbf{C}^{job}$ to all its idle neighbors. The agent receives the favor-granted messages (i.e. $\mathbf{C}^{fav}$) from all robots, except Robot O1. As there are more than one neighbors who fit the criteria, there are multiple available solutions. From the scores in Fig. 4.19, we can observe that Robot S1 has the highest score. Results can be verified with the visualized capability in Fig. 4.14, where Robot S1 has the most similar capability to that of Job 2. Thus, the job is shared with Robot S1 in a Partnership form of collaboration.

Active Distributed Methodology

Job 2 requires a type 1 manipulation component. However, Robot S2 has a type 2 manipulation component, which means its capabilities do not match the requirement. As the solo mode is not feasible due to the lack of a type 1 manipulation component, Robot S2 needs a helper. A help request is broadcasted to the idle neighbors. The agent receives the capabilities of all idle neighbors. As the solo mode is not feasible and there are insufficient assembly ports on Robot S2, based on Fig. 4.1, the remaining solution is to adopt the Subcontract mode. There are 4 potential Subcontract collaboration solutions. The optimal solution is located via the scoring calculation where Robot S1 is the optimal choice with the highest score $\hat{\theta} = 50$ (see Fig. 4.19). Visually, Robot S1's color is the most similar to that of Job 2 shown in Fig. 4.14. Hence, the job is transferred to Robot S1 with the Subcontract mode.

We also compare the performance differences via experiments. In Fig. 4.18(c), we assign the same Job 1 to Robot M1 to evaluate the results differences and accuracy in selecting a member under passive and active approaches. Both approaches show the same results, with Robot O1 selected as the optimal neighbor.

Both methodologies are valid and have their own features. The passive approach puts individual consideration at the top priority while the active approach puts more emphasis on resource allocation. The differences in practical scenarios are that the passive strategy is more suitable when resources are not limited. The active strategy is more applicable in situations where resources are scarce. More details are discussed in the Sec. 4.8.1.

4.7.7 ODKG Validation

We validate the ODKG model by assigning various types of jobs to our heterogeneous MRS. When the robot receives a job assignment, the agent builds the ODKG model and uses it for decision-making. In Fig. 4.12 and 4.13, the teams are capable of completing the assignment in solo or working collaboratively. Consider the three-robot assembly case with the use of Method 2 in the following (see Fig. 4.15). After the assembly, three robot ontology layers are connected. In the beginning, all objects 'box' are linked to the object 'table'. The locomotion capability node 'L' of the orange robot which is denoted as worker 'C' in ODKG and 'table' are connected with the object 'ground' as shown in Fig. 4.15 (5). When the purple robot (denoted as 'A' in ODKG) sees the 'box', there exists an interaction between the 'box' and the perception node 'P' of the worker 'A', demonstrated in Fig. 4.15 (6). As the kinematic graph shows that the observed 'box' is on the 'table', the robot has to go to the 'table' to reach the 'box'. Thus, the robot team moves towards the 'table' with the locomotion capability of 'C' and grasps the 'box' with the manipulation capability of 'A' in Fig. 4.15 (7). The team locates and moves to the target 'place' for unloading the 'box' shown in Fig. 4.15 (8)–(9). This process is repeated until all boxes are linked to the 'place' as illustrated in Fig. 4.15 (10). In this numerical example, the agent formulates the collaboration strategy and completes the job successfully.

Both passive and active methods provide similar results in the experiments. In the experiments shown in Fig. 4.18 (a) with the Job 1 Method 1 and 2, the agent Robot M collaborates with its member Robot O in assembly mode. An ODKG is constructed and evolves with time. From steps 1 to 3 shown in Fig. 4.18 (a), the unique perception information gathered by the Robot O (i.e. the location of the non-explosive goods) is shared to Robot M. The agent then triggers the 'move' and 'grasp' actions of its 'locomotion' and 'manipulation' capabilities with the proposed hardware coordination system.

We tested the partnership and solo modes with different approaches in the Job 2 experiments (see Fig. 4.18 (a)). The agent Robot M2 receives the job and works with Robot M1 in partnership mode. If Robot M1 lacks sufficient power, then Robot M2 works alone in solo mode. We use the robot layer in ODKG to coordinate the hardware allocation problem. In partnership mode, the least cost path from a feedback node (e.g. 'perception' node) to any action-related node (e.g. 'locomotion', 'manipulation' capability node) is 2. It indicates that using the node from the same robot which receives information is the best hardware allocation result.

In the experiments with Job 3 Method 1 and 2, the agent Robot L receives the job and collaborates with Robot M. An ODKG is built and used to coordinate the hardware resources in the team. As both robots are mobile, Robot M sends a 'move' action command to the ODKG to decide which robot's component to be used when it locates the goods in Step 3 shown in Fig. 4.18 (a). However, after reaching and grasping the goods, Robot M loses sight. The team relies on the perception information from the agent and uses the agent's 'locomotion' capability to move towards the unloading area. When the team reaches the unloading area, the team uses the 'manipulation' capability from Robot M to release the goods.

To further evaluate the effectiveness of the ODKG application, we introduce Robot O4 which is a Type O4 robot. We create a new job: Job 5, which requires an 8-unit of height, type 1 and 3 perception capabilities to determine the 'non-simulated explosive' box, and type 1 manipulation and locomotion components. In Fig. 4.18 (b), the agent Robot L finds the correct robots to be its helper: Robot L assembles with Robot M1 and Robot O1. During the task execution, they use the type 3 'perception' device of Robot O1 to detect hot objects. By combining the perception data of Robot O1 with Robot M1, the team successfully locates the non-hot box. With the hardware optimization in ODKG, the team uses the ‘manipulation’ component of Robot M1 to interact with the non-hot box and control the ‘locomotion’ of Robot L to navigate the environment. We illustrate the performance of the proposed methods in the accompanying video https://vimeo.com/726691079

4.8 Discussion and Summary

4.8.1 Discussion

We evaluate the proposed ODKG with different working modes and collaboration strategies. Through simulations and experiments, we demonstrate the efficiency of ODKG in heterogeneous MRS. The average task efficiency of adopting various working modes is shown in 4.16 and 4.18 (d). Both results proved that the partnership mode gives the optimal outcome with the shortest elapsed time for job accomplishment.

Both approaches are feasible and effective in capability-based task allocation. The passive approach allows neighbors to share some computational cost from the agent by focusing on individual offerings, but not the whole group. The active approach however values the resource utilization of the whole community before the individual. Passive and active methods are related to the collaboration attitude while strengths and types are for the capability modelling. These methods can be used interchangeably. For example, the passive approach can be used with type-based modelling and vice versa. In Method 1, the average elapsed time of ODKG (Phase 1) from job receives to member selection is around 0.17s in both small (8 robots) and large-scale (24 robots) simulation environments. Method 2 takes around 0.18s and 2.53s respectively as the agent requires more time to compare all neighbors' capabilities with the requirement, thus, the time increases with the number of neighbors. The system performance in terms of member selection time remains consistent when using Method 1, regardless of the number of robots involved. The selection time will stay similar even with a small number of robots or a significantly larger deployment, such as 100 or 1000 robots. In contrast, the selection time increases with Method 2 as the number of robots grows.

There are two main limitations of the proposed methodology: (1) the processing time increases with the number of capabilities, strength levels, capability types, and neighbors, and (2) the capabilities of all robots have to be quantified beforehand. Various kinds of jobs can be assigned to the robots to evaluate their strength level and to classify them into different categories.

4.8.2 Summary

This chapter focuses on the capability sharing in heterogeneous multi-robot collaboration. Specifically, it presents a distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous MRS. Our methodology is designed to handle various kinds of working modes/collaboration between heterogeneous robots and to broaden the operability scope of the job nature of a robot. Two approaches are presented and their performances are tested through numerical simulation and real-world experiments. Various multi-robot collaborative tasks guided by the proposed method are conducted with optimized strategies. Future work aims to extend our proposed method to include various self-reconfigurable robots. Different real-world applications and collaboration modes will be explored to validate our method's performance.

Chapter 5

Capability Enhancement: Robot Collaborative Manipulation

5.1 Introduction

In addition to amplifying and extending a robot's existing capabilities, the ability to utilize tools provides greater potential for manipulation approaches, thereby enhancing the robot's dexterity in interacting with its environment. In other words, tool use can be viewed as a further extension of the robot's current configuration, leading to an increase in dexterity and an enhancement of its initial manipulation capabilities. To allows the robot to perform tool manipulation task based on the task instruction expressed by human, it involves the ability to understand task instructions expressed in free-form language and to generate appropriate motion plans. While various models have been developed using Large Language Models (LLMs), the integration of LLM-based tool manipulation with non-prehensile actions remains an open and under-explored topic.

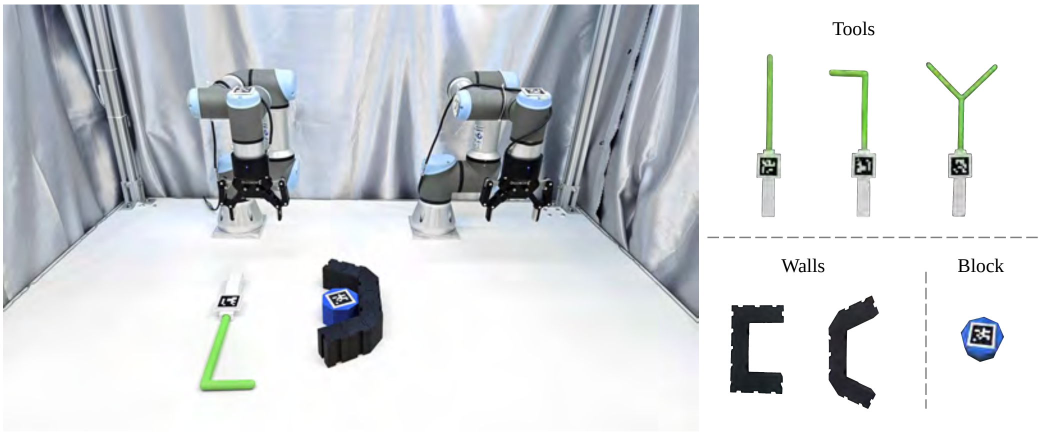
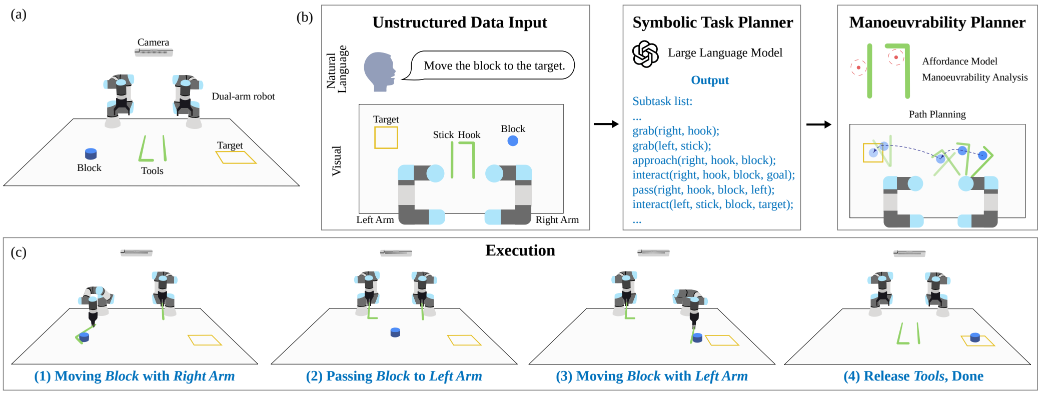


Figure 5.1: Tool-Object manipulation in a dual-arm robotics system with environmental constraints using the non-prehensile approach.

This chapter focuses on the capability enhancement in robot collaborative manipulation aspect. Specifically, we aims at enhancing the manipulation capability of a dual-arm robot via tool usage. We propose an novel LLM-based manoeuvrability-driven method with the following original contributions: (1) We develop an effective model to represent the geometric-mechanical relations and manoeuvrability of tools and objects; (2) We propose a non-prehensile strategy to manoeuvre objects under different constraints with tools; (3) We evaluate the performance of the proposed methodology with real-world experiments on a dual-arm robotic system.

The rest of the chapter is organised as follows: Sec. 5.2 presents the methodology, Sec. 5.3 presents the results, Sec. 5.4 gives final conclusions.



```
Subtask list:
...
grab(right, hook);
grab(left, stick);
approach(right, hook, block);
interact(right, hook, block, goal);
pass(right, hook, block, left);
interact(left, stick, block, target);
...
```

Figure 5.2: (a) The task environment includes a camera for real-time top-view capturing, a dual-arm robot, tool(s), and a blue block to be manipulated to the target location. (b) The architecture of our system: Unstructured data input is converted to a subtask list in the symbolic task planner with an LLM, a manoeuvrability-driven planner to compute the tool's manoeuvrability and generate an affordance-oriented motion and path. (c) Execution process of the result given by the system: dual-arm robots take turns pushing the blue block from one side to another via collaboration.

5.2 Methodology

5.2.1 Problem Formulation

Consider a dual-arm robotic system using a tool to manipulate a block at a far distance (see Fig. 5.1). Given the input is a free-form language task $\mathbf{L}$ (e.g., "move the block to Point B"), we apply a high-level symbolic planner (i.e., a LLM) to decompose the task into multiple subtasks $\mathbf{l}_i$, $\mathbf{L} = \{\mathbf{l}_1, \mathbf{l}_2, \dots\}$ where $\mathbf{L}$ contains a list of pre-defined motion functions $\mathbf{l}_i$.

We define a *tool* as a manipulable object that is graspable by a robot, a *manipulandum* [90] as an object (e.g. a block) that is manipulated via a tool, and a *wall* as a static non-manipulable object. Tool use by robots is challenging as the tools can have various shapes, the environment can be dynamic, and the contact between the tool and the manipulandum may be hard to maintain in a long-horizon task. Depending on the geometric features of a tool and a wall, the available affordance for manoeuvring a manipulandum may be different. Affordance here refers to the available action-effects offered by the tool or the environment. In this work, we classify affordance into two types: active and passive. Active affordance is given from a manipulable object, i.e. a tool, and it is directly related to the manoeuvrability when driving a manipulandum. Passive affordance is given from a static non-manipulable object.

To derive our methodology, the following setup assumptions are made: (1) The manipulation motion is planar, and (2) the size of the manipulandum is not larger than any one of the segments of the tool. Throughout this chapter, we denote "tool-based object manipulation" as TOM, and "tool-based object manipulation under environmental constraints" as TOME. Also, we use $\mathbf{p}^{\circ}$ to represent the 2D pose of an object $\circ$. The complete architecture of our method is depicted in Fig. 5.2.

5.2.2 LLM-Based High-Level Symbolic Task Planner

To obtain a valid task decomposition for a long-horizon task, the system needs to understand the requirements and generate an executable subtask list. We develop a symbolic task planner that takes natural language instructions with scene descriptions as input, and outputs a list of high-level subtasks. The list involves the tool selection/sharing between two arms, the sequence to manipulate the tool with the manipulandum, and the interaction between the two arms. The model is fine-tuned with around 20,000 lists of example data. These two data sets are generated randomly with different environment settings, such as different locations of the manipulandum, target destination, robots, and tool shapes.

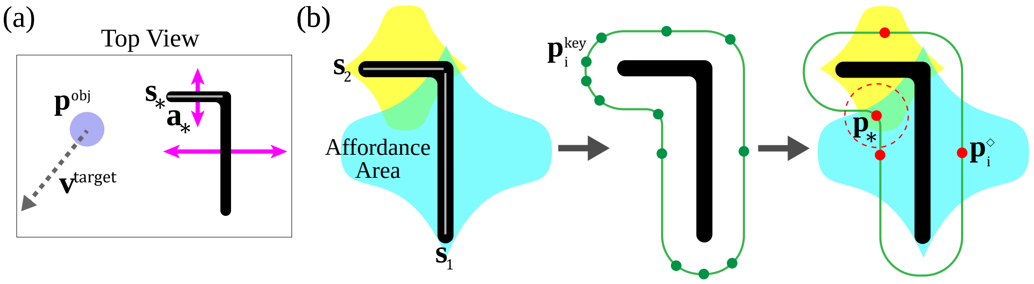
The system interprets the provided high-level task $\mathbf{L}$, which can have a structure like “Please move the blue block to the right-hand side”, “Can you push the block to the target?”, etc. Visual information of the scene is grounded to the system from the observation data $\mathbf{o}$, where $\mathbf{o}$ is composed of a series of data points, such as the pose of the block (manipulandum), tools, robots, and walls. The system embeds the environmental information with the task instruction to produce a desired configuration requirement, denoted as $\{\mathbf{p}^{\text{obj}}, \mathbf{p}^{\text{target}}, \dots\} \leftarrow f(\mathbf{L}, \mathbf{o})$ where $f(\mathbf{L}, \mathbf{o})$ is the embedded result.

The LLM interprets the output of $f(\mathbf{L}, \mathbf{o})$ to generate a subtask list $\{\mathbf{l}_1, \mathbf{l}_2, \dots\} \leftarrow f_{\text{llm}}(f(\mathbf{L}, \mathbf{o}))$ where $\mathbf{l}_i$ is a subtask describing the manipulation phase of each robot [114] and is corresponding to a high-level robot motion function. The motion functions are designed to be simple and specify a short-term goal of the concerned object (these functions omit low-level motion commands). For simplicity, here we use m to represent manipulandum in the following function definitions. We use $\texttt{grab}(arm, tool)$ for grabbing a *tool* with the robot *arm*; $\texttt{approach}(arm, tool, m)$ for approaching the location of m with *tool* using *arm*; $\texttt{interact}(arm, tool, m, goal)$ for moving m to the *goal* location with the *tool*; $\texttt{stepping}(arm, tool, m)$ for moving m out from the bounded are with the *tool* of the *arm* through contact pulsing motions; $\texttt{pass}(arm1, tool, m, arm2)$ for passing m to another arm’s workspace; $\texttt{release}(arm, tool)$ for releasing the *tool* back to its original place with the *arm*.

A sample motion task with a dual-arm robot can given as:

$$\texttt{pass}(right, hook, block, left);$$

$$\texttt{approach}(left, stick, block);$$



(a)
Top View
$\mathbf{p}^{\text{obj}}$
$\mathbf{v}^{\text{target}}$
$\mathbf{s}_*$
$\mathbf{a}_*$
(b)
$\mathbf{s}_2$
$\mathbf{s}_1$
Affordance Area
$\mathbf{p}_i^{\text{key}}$
$\mathbf{p}_*$
$\mathbf{p}_i^{\diamond}$

Figure 5.3: (a) Affordance vectors are shown in pink arrows. Grey arrow is $\mathbf{v}^{\text{target}}$ and the desired affordance vector is denoted as $\mathbf{a}_*$. (b) shows the manoeuvrability analysis flowchart: affordance area is visualised with the Gaussian function in yellow and blue; expand and downsample the tool's shape to get key points $\mathbf{P}^{key}$ (green colour dots); combine the affordance area with the key points $\mathbf{P}^{key}$ to get the non-redundant points $\mathbf{P}^{\diamond}$ (red dots), and combine the affordance $\mathbf{a}_*$ found in (a) to obtain the position for the manipulandum to be at with the tool (labelled as $\mathbf{p}_*$ with a red dot) and the highest manoeuvrability region is shown with a dashed red circle.

$$\texttt{interact}(left, stick, block, target); \cdots \leftarrow f_{\text{llm}}(f(\mathbf{L}, \mathbf{o}))$$

where both arms take turns manipulating the block. The right arm passes the block to the left by pushing it to an area where both arms can reach it. The left arm approaches the block with a stick and manipulates the block to the target. To this end, the symbolic task planner converts the unstructured data to a series of motion functions, including robot motion, tool planning, manipulation sequence, and collaboration.

5.2.3 Visual Affordance Model

Tools can have various shapes and complex structures. In this chapter, we focus on the following tool geometries: a stick, an L-shaped hook, and a Y-shaped hook. Affordances are related to the geometric features of a tool [115]. To analyse the possible affordances, we divide the tool into smaller segments (i.e. a line), and denote them as $\mathbf{S} = \{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ where $\mathbf{s}_i$ and $\mathbf{s}_{i+1}$ are segments next to each other. We compute the normal vectors of the segment at the middle point and scale them by half of the segment's length. This is done to weight the affordance effect this regions carries. There are two affordance vectors per segment $\mathbf{s}_i$, each pointing in opposite directions, as depicted in Fig. 5.3(a). Let us define $\mathbf{A} = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{2n}\}$ as the structure that contains all the affordance vectors $\mathbf{a}_i$, for n as the number of segments.

To determine which affordance vector $\mathbf{a}_i$ will be used to interact with the manipulandum, we compare the similarity between $\mathbf{a}_i$ and the vector from the manipulandum's position to the target point $\mathbf{v}^{\text{target}}$ by:

$$\theta_i = \cos^{-1}\left(\frac{\mathbf{v}^{\text{target}} \cdot \mathbf{a}_i}{||\mathbf{v}^{\text{target}}||\,||\mathbf{a}_i||}\right) \tag{5.1}$$

where θ_i is the similarity score. The optimal affordance vector $\mathbf{a}_*$ and its according segment $\mathbf{s}_*$ are found by locating the vector that has the minimum similarity score $\arg\min_{\mathbf{a}}(\Theta)$ where $\Theta = \{\theta_1, \theta_2, \ldots\}$.

5.2.4 Manoeuvrability Analysis

A tool can push the manipulandum from the side, from the tip, or other areas. However, the relative location of the manipulandum respective to the tool affects its manoeuvrability. In other words, the affordance provided by the tool is proportional to manoeuvrability. Consider using a rotating stick to push an object with its end tip. In this situation, the tool may lose contact with the manipulandum as it rolls outwards, hence, the manoeuvrability of this point is low. On the other hand, the midpoint of the stick has a high manoeuvrability, which proportionally decreases as the contact point is further away from the midpoint. We can model this behaviour with a Gaussian function, where its centre is the segment's centre and the peak height is half the segment's length, see Fig. 5.3(b). We refer to this region as an affordance area.

All the pixels in the affordance area of $\mathbf{s}_i$ are set to 1 in an image frame $\mathbf{I}_i$ and the rest to 0, which creates a binary image; This process is repeated for all segments. All binary images are then summed as $\hat{\mathbf{I}} = \sum_{i=1}^{n} \mathbf{I}_i$ where n is the number of segments. The affordance of tool segment is quantified with the (normalised) manoeuvrability matrix: $\mathbf{M} = \hat{\mathbf{I}}/\hat{I}_{max}$, for $\hat{I}_{max}$ as the maximum value in $\hat{\mathbf{I}}$. Tool regions with high values in the image $\mathbf{M}$ reflect a high manoeuvrability.

These computed manoeuvrability values are useful to determine the location where the tool interacts with the manipulandum. To determine the centre of the object, we then expand the contour of the tool by the object's radius r^{obj}. This contour is downsamppled with the Ramer-Douglas-Peucker algorithm [116], then, parameterised with the spline fitting technique reported in [117]. To extract key features of the tool geometry, we use a sliding window strategy to examine a small number of neighbouring points. If there exists a point where its curvature is larger than a threshold in the local neighbourhood, we consider this point as one of the feature points.

To compute the minimal number of key points which we denoted as follows:

$$\mathbf{P}^{key} = \{\mathbf{p}_1^{key}, \mathbf{p}_2^{key}, \dots\}$$

that capture the highest manoeuvrability among feature points, we use the density-based clustering algorithm from [118]. By integrating the affordance areas we obtained earlier, we can filter out some redundant key points. For example, if

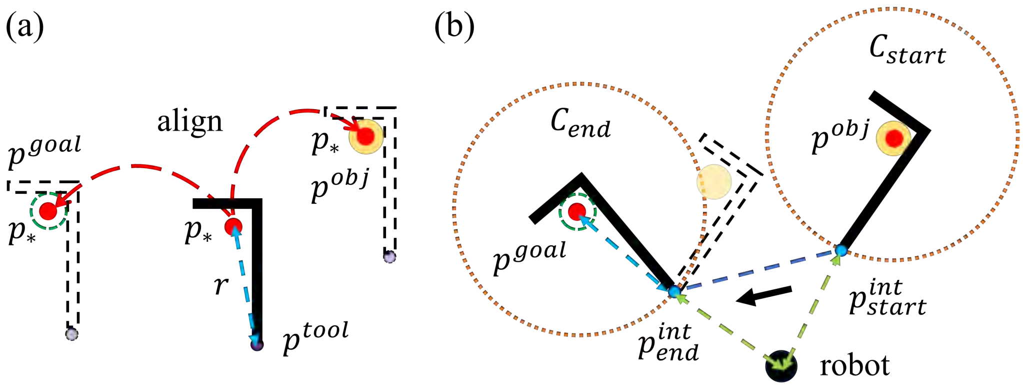


Figure 5.4: (a) The tool is virtually aligned to the current object and the goal location, with $\mathbf{p}_* = \mathbf{p}^{obj}$ and $\mathbf{p}_* = \mathbf{p}^{goal}$. (b) The light blue dashed line is the radius of the orange circle $\mathbf{C}_{\text{start}}$ and $\mathbf{C}_{\text{end}}$, which equals the distance between $\mathbf{p}^{tool}$ and $\mathbf{p}_*$. The tool moves from $\mathbf{p}^{\text{int}}_{\text{start}}$ to $\mathbf{p}^{\text{int}}_{\text{end}}$ by following the dark blue dashed trajectory line.

there exists a point $\mathbf{p}_i^{key}$ located outside the affordance area (visualised in Fig. 5.3(b)), we consider this point as redundant. All the non-redundant points are then grouped into $\mathbf{P}^\diamond = \{\mathbf{p}_1^\diamond, \mathbf{p}_2^\diamond, \dots\}$. To find the point in $\mathbf{P}^\diamond$ with the highest manoeuvrability (defined as $\mathbf{p}_*$), we use the manoeuvrability matrix $\mathbf{M}$ and distance between $\mathbf{p}_i^\diamond$ and $\mathbf{a}_*$ as described in the metric below:

$$\mathbf{p}_* = \arg\min_{\mathbf{p}_i^\diamond}((1 - [\mathbf{M}]_{\mathbf{p}_i^\diamond}) + \|\mathbf{p}_i^\diamond - \mathbf{a}_*\|) \tag{5.2}$$

where $[\mathbf{M}]_{\mathbf{p}_i^\diamond}$ denotes to the image value of $\mathbf{M}$ at point $\mathbf{p}_i^\diamond$. The region with the highest manoeuvrability is defined as the circle (with object radius) centred at $\mathbf{p}_*$. (see Fig. 5.3(b))

5.2.5 Manoeuvrability-Oriented Controller

The subtask “`interact`” triggers the robot to use the selected tool to drive the manipulandum towards the desired location. In this section, we derive our method to perform this type of motion assuming that the tool approaches the object and is going to make contact with it in the subtask “`interact`”.

Initial and Final Poses

The tool’s pose corresponds to its grasping configuration, which coincides with the robot end-effector’s pose when the robot grasps the tool (see Fig. 5.4). We use $\mathbf{p}^{\text{tool}}$ to denote the tool’s grasping point (x, y coordinates) when it has not come in contact with the object. To construct a trajectory for tool-based object transport, we need to find out the tool’s desired initial and final poses for the subtask “`interact`”. We first define these poses (which include the orientation) of the chosen tool as $\mathbf{p}^{\text{int}}_{\text{start}}$ and $\mathbf{p}^{\text{int}}_{\text{end}}$ respectively.

To efficiently move the object, we propose a method that reduces the travel distance while ensuring continuous contact. In the first contact, we align the highest manoeuvrability point $\mathbf{p}_*$ of the tool to the object’s centre $\mathbf{p}^{\text{obj}}$, where $\mathbf{p}_* = \mathbf{p}^{\text{obj}}$.

The motion trajectory of a tool, moving along the z-axis of the object’s centre without displacing it can be described as a circular trajectory with the centre $\mathbf{p}^{\text{obj}}$ and radius r, where $r = \|\mathbf{p}_* - \mathbf{p}^{\text{tool}}\|$. We represent the trajectories for the initial and final configurations as $\mathbf{C}_{\text{start}}$ and $\mathbf{C}_{\text{end}}$ (see Fig. 5.4(a)).

The possible location for $\mathbf{p}^{\text{int,x,y}}_{\text{start}}$ will be lying on $\mathbf{C}_{\text{start}}$ and can be determined by finding a point on $\mathbf{C}_{\text{start}}$ which it is the closest point to the robot (the distance is indicated with a light green dashed line in Fig. 5.4(b)). Based on the tool’s

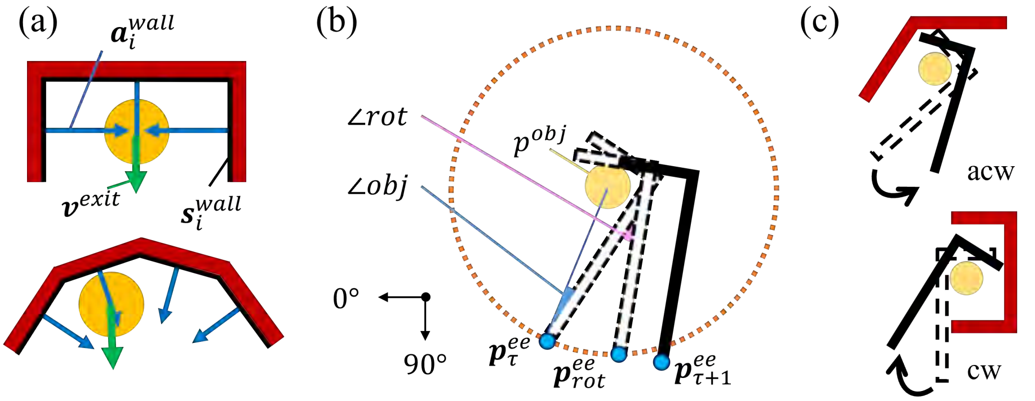


Figure 5.5: (a) Walls are in red with the segment of the wall $\mathbf{s}_i^{\text{wall}}$ highlighted in black; blue arrows are the passive affordance vector and green arrows indicate the moving direction of $\mathbf{v}^{\text{exit}}$. (b) The tool pose moves from τ to $\tau + 1$ by rotating with $\angle$rot and translating linearly to $\mathbf{p}_{\tau+1}^{\text{ee}}$. (c) Rotation direction of a tool: anti-clockwise and clockwise direction.

geometry, we can determine the orientation of the initial pose $\mathbf{p}_{\text{start}}^{\text{int}}$; The same approach applies to $\mathbf{p}_{\text{end}}^{\text{int}}$.

Motion Strategy

To stably move from $\mathbf{p}_{\text{start}}^{\text{int}}$ to $\mathbf{p}_{\text{end}}^{\text{int}}$, the following motion strategy is implemented to achieve the task: First, the robot aligns $\mathbf{p}_*$ with $\mathbf{p}^{\text{obj}}$ and matches $\mathbf{p}^{\text{tool}}$ with $\mathbf{p}_{\text{start}}^{\text{int}}$; then translates along the x and y axes until it reaches $\mathbf{p}_{\text{end}}^{\text{int,x,y}}$; lastly, the tool is rotated to align with the orientation of $\mathbf{p}_{\text{end}}^{\text{int}}$.

5.2.6 Application with Environmental Constraints

When moving an object across a table, we may encounter constraints from the environment, such as walls. These constraints restrict the potential movement directions of the object. Formally, a constrained area can be defined by a series of points where more than one axis of freedom of the manipulandum motion may be restricted. In this section, we focus on the motion triggered by the subtask '`stepping`'.

Consider the manipulandum is tightly confined within a concave-shaped wall, as shown in Fig. 5.5(a), with an unknown exit and assume that the tool can enter the constrained area. To move the manipulandum out from the bounded area with small movement space, we determine the direction from the manipulandum to the exit by considering the overall affordance of the wall boundary. We denote this direction vector as $\mathbf{v}^{\text{exit}}$, and its magnitude is defined as the minimum travel distance for the manipulandum. We consider the inner edge of the wall as a segment $\mathbf{s}_i^{\text{wall}}$ where $i = \{1, \ldots, n^{\text{wall}}\}$ and n^{wall} is the number of the wall segment. The affordance of a wall is passively provided and is defined as $\mathbf{a}_i^{\text{wall}}$ with the model shown in Sec. 5.2.3. The passive affordance vector is the normal vector of $\mathbf{s}_i^{\text{wall}}$ located in the middle, and the magnitude is scaled to half of $\mathbf{s}_i^{\text{wall}}$ with the direction pointing towards the constrained area. The moving direction for the manipulandum to the exit can be obtained by: $\mathbf{v}^{\text{exit}} = \sum_{i=1}^{n^{\text{wall}}} \mathbf{a}_i^{\text{wall}} + \mathbf{p}^{\text{obj}}$ where $\mathbf{v}^{\text{exit}}$ integrates all passive wall affordance vectors, see 5.5(a).

Given that only part of the tool can enter the confined area, our primary focus is the tip of the tool. The segment connecting of the tool's tip is denoted as $\mathbf{s}^{\text{tip}}$, with its corresponding affordance vector denoted as $\mathbf{a}^{\text{tip}}$. The desired rotation angle of the end pose of $\mathbf{a}^{\text{tip}}$ is the angle of $\mathbf{v}^{\text{exit}}$.

The highest manoeuvrability region can be obtained by treating $\mathbf{v}^{\text{exit}}$ as the target vector $\mathbf{v}^{\text{target}}$, $\mathbf{a}^{\text{tip}}$ as the desired affordance $\mathbf{a}_*$, and assuming the tool is rotated such that $\mathbf{a}^{\text{tip}} = b\mathbf{v}^{\text{exit}}$ with $b > 0$ as a scaling factor. We first align $\mathbf{s}^{\text{tip}}$ to the first segment of the wall (i.e. $\mathbf{s}_1$), with $\mathbf{p}^{\text{obj}}$ inside the highest manoeuvrability region of the tool. The tool approaches the object and maintains contact with the manipulandum by minimising the distance $\|\mathbf{p}_* - \mathbf{p}^{\text{obj}}\|$.

To move in the limited area while interacting with the block, we employ a stepping approach to manipulate the block in the confined area. As the possible movement area is small and highly restricted, an incremental pulsing motion is adopted to make small adjustments with high accuracy motion control to the tool and the manipulandum. Inspired by the animal manipulation study in [85] (where a crow uses a tool to get the food from the box slot by rotating and dragging the tool outwards), we adopt a similar approach to retrieve the object from confined spaces. This strategy continuously alternates between "repositioning" the tool and incremental "rotation-dragging" the object towards the exit until it can be fully extracted. This strategy is illustrated in Fig. 5.5.

We define "repositioning" as moving the tool closer to the object and realigning $\mathbf{p}_*$ with $\mathbf{p}^{\text{obj}}$ by k amount. In "rotation-dragging", the tool maintains contact with the manipulandum when it rotates by a certain angle as $\angle$rot shown in Fig. 5.5(b) and moves outwards by extending $\overrightarrow{\mathbf{p}^{\text{ee}}_{\tau}\mathbf{p}^{\text{ee}}_{\text{rot}}}$ by a $w > 0$ amount.

We define τ as an action step variable and is incremented by 1 if an action (reposition/rotation-dragging) is fulfilled (i.e. $\tau = 0, 1, 2, \ldots$). To control the change of action, a step function (denoted as $u(\tau)$) is implemented as a trigger with the step variable τ. This kind of non-prehensile crow-inspired behaviour can be unified and modelled as:

$$\mathbf{p}^{\text{ee}}_{\tau+1} = \begin{bmatrix} \mathbf{p}^{\text{ee},x}_{\tau} \\ \mathbf{p}^{\text{ee},y}_{\tau} \\ \phi_{\tau} \end{bmatrix} + u(\tau) \begin{bmatrix} k(\mathbf{p}^{obj,x}_{\tau} - \mathbf{p}^{x}_{*}) \\ k(\mathbf{p}^{obj,y}_{\tau} - \mathbf{p}^{y}_{*}) \\ 0 \end{bmatrix}$$

$$+ u(\tau+1) \begin{bmatrix} w(\mathbf{p}^{\text{obj},x}_{\tau} - r\cos(\phi_{\tau}) - \mathbf{p}^{\text{ee},x}_{\tau}) \\ w(\mathbf{p}^{\text{obj},y}_{\tau} + r\sin(\phi_{\tau}) - \mathbf{p}^{\text{ee},y}_{\tau}) \\ f(\phi_{\tau+1}) \end{bmatrix}$$

$$u(\tau) = \begin{cases} 0, & \text{if } \tau \text{ is odd} \\ 1, & \text{if } \tau \text{ is even} \end{cases} \tag{5.3}$$

where $\mathbf{p}^{\text{ee}}_{\tau+1}$ is the next target pose of the end-effector at the action step $\tau+1$ for the affordance vector $\mathbf{a}^{\text{tip}}$ not parallel to $\mathbf{v}^{\text{exit}}$, such that $\mathbf{a}^{\text{tip}} \neq b\mathbf{v}^{\text{exit}}$. The angle of the tool at $\tau+1$ (denoted as $\phi_{\tau+1}$) depends on the rotational direction (see Fig. 5.5), that $\phi_{\tau+1}$ is computed as

$$f(\phi_{\tau+1}) = \begin{cases} -\angle\text{obj} - \angle\text{rot} & \text{if direction is anti-clockwise} \\ -\phi_{\tau} + \pi - \angle\text{obj} - \angle\text{rot} & \text{otherwise} \end{cases} \tag{5.4}$$

where ϕ_{τ} is the tool's angle at the action step τ, $\angle$obj is the angle between the block, grasping point, and a tool's keypoint, $\angle$rot is the amount of angle to rotate.

5.3 Results

To validate our methodology in terms of accuracy and robustness, we have conducted around 200 experiments in a dual-arm robot system. In the experiment, two sets of UR-3 robotic arms are used and GPT 3.5 is implemented for task decomposition. Three types of tools are selected which are a stick, an L-shaped hook, and a Y-shaped hook (see Fig.5.1). Different tool combinations are evaluated with diverse movement directions and tasks. A RealSense D415 captures the images of the whole process. Data is passed to a Linux-based computer with the Robot Operating System (ROS) for image process and robot control. Aruco code is used for providing accurate pose tracing in real time.

These experiments include validating the task decomposition performance in a single and dual-arm robot setup, the robustness of the affordance and manoeuvrability model in various shapes of tools, and evaluating the overall performance.

5.3.1 Single-Arm Robot with a Single Tool

We first evaluate the task decomposition performance of LLM. For that, a tool and a blue block are placed on the table with the target given as shown in Fig. 5.6. The task is to manipulate the block within a close distance, which is sufficient for a single-arm robot. The embedded information which contains the task, the environment and the geometry of the tool is passed to the LLM. In the experiment shown in Fig. 5.6(a), the robot executes the subtasks generated by the high-level symbolic task planner which include: $\texttt{grab}(right, hook)$; $\texttt{approach}(right, hook, block)$; $\texttt{interact}(right, hook, block, target)$;$\texttt{release}(right, hook)$, where the right arm first moves and grabs the hook, then moves the block to the target, and lastly releases the tool back to its original place. The experiment showcases the application of the proposed affordance and manoeuvrability model in locating the highest manoeuvrability region for block transportation. During the manipulation stage, the block is kept within the highest manoeuvrability region (indicated with a red circle in Fig. 5.6) to receive affordance effectively from the tool. The minimisation of the error between the $\mathbf{p}^{\text{obj}}$ and the $\mathbf{p}^{\text{target}}$ for each experiment is shown in Fig. 5.7. These results corroborate that the proposed method can be used to actively drive a robot to manipulate an object via a tool.

5.3.2 Dual-Arm Robot with Long-Horizon Task

We then evaluate the long-horizon task performance where the block has to travel from far right to far left, far right/left to top right/left, and vice versa. The long-horizon task is evaluated with multiple tool combinations. The system observes and generates a collaborative motion plan. In the experiment shown in Fig. 5.8(a), the right and left arms pick up the stick and the hook respectively. The right arm uses the stick to push the block to the left side, allowing the left arm to continue the task. The robot leverages the advantage of the hook to drag the block closer to its working area and push the block to the desired location. In Fig. 5.8(b), the right and left arms grasped the Y-shaped tool and the stick respectively. The right arm uses the tool to pass the block to the left. The left arm uses the stick to push the block to the target location.

The long-horizon task performance is evaluated with the tool-sharing ability. Assuming there is only one tool available, it has to be shared among the dual-arm robot. Fig. 5.8(c) demonstrates the tool is passed to another arm once the block

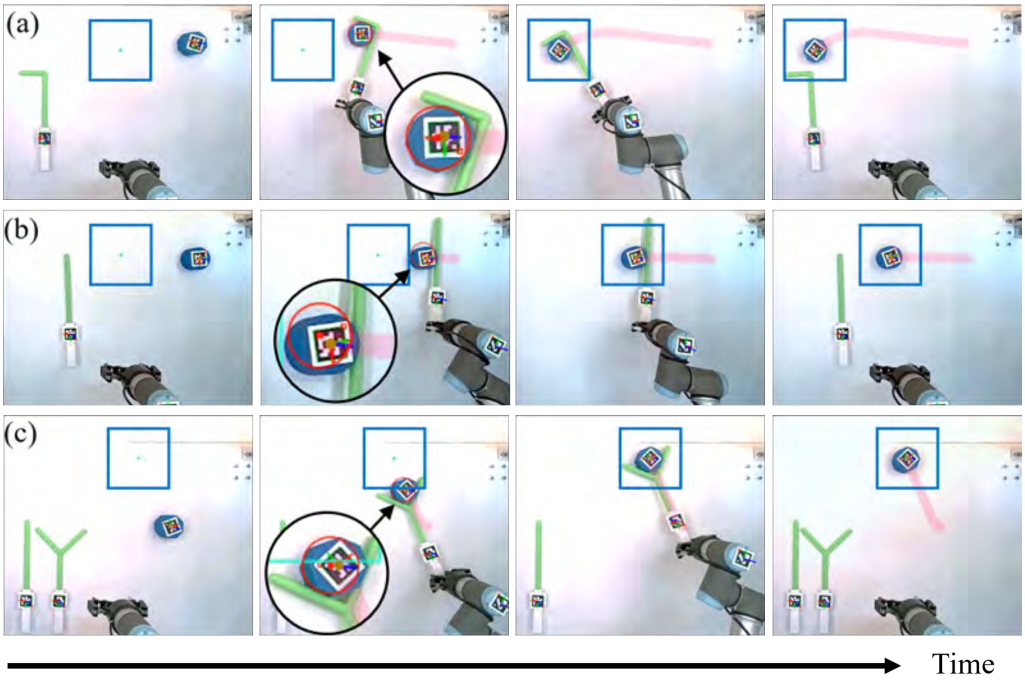


Figure 5.6: Single-arm robot with a single tool: moving the block from (a) right to left with a hook, (b) right to left with a stick, (c) bottom to top with a Y-shaped tool. The trajectory of the block is reflected in the red line. The highest manoeuvrability point is indicated with a red circle.

is pushed to the middle of the table. The block is moved accurately to the target with motion-decomposed: ‘`grab`; `approach`; `interact`; `pass`; `release`; `grab`; `approach`; `interact`; `release`’ where the left arm releases the tool once it is done and the right picks up the tool to continue moving the block. Though the hook is in a two-link geometry, the pushing is afforded by the right side of the tool (a single segment) with the highest manoeuvrability region.

The minimisation of the error between $\mathbf{p}^{\text{obj}}$ and $\mathbf{p}^{\text{target}}$ for each experiment is shown in Fig. 5.9. Similar to the single-arm robot with a single tool experiment, this long-horizon task experiment also demonstrates the robustness of the proposed methodology such that the tasks are successfully decomposed into mul-

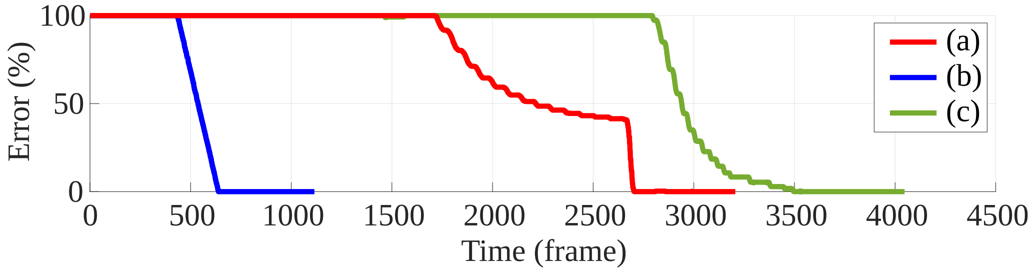


Figure 5.7: Evolution of the minimisation process of the error between the current object position and the target for the tasks shown in Fig. 5.6.

tiple collaborative subtasks, and the highest manoeuvrability region of the tool is leveraged in block manipulation.

5.3.3 Tool-Object Manipulation in Constrained Environments

To further evaluate the performance of the model in application scenarios, different shapes of walls are constructed as shown in Fig. 5.8(d)–(e). Two walls are designed with 90-degree and 65-degree for the inner-angles. Maneuvering a hook within a confined space presents greater challenges compared to using a stick. Additionally, a Y-shaped hook proves unsuitable for dragging objects in tight quarters. Therefore, in this experimental study, we opt for a hook tool with a right arm to navigate effectively within the constrained environment. Similar to the previous results, Fig. 5.8(d)–(e) also implements the task planner successfully to decompose the task and applies the stepping controller for object manipulation. The tool first aligns its $\mathbf{s}^{\text{tip}}$ to the first segment of the wall and adopts the proposed non-prehensile stepping motion controller stated in (5.3). The block is dragged out from the confined area by alternating between the action of ‘repositioning’ and ‘rotation-dragging’.

During the pulsing manipulation, the block maintains contact with the highest

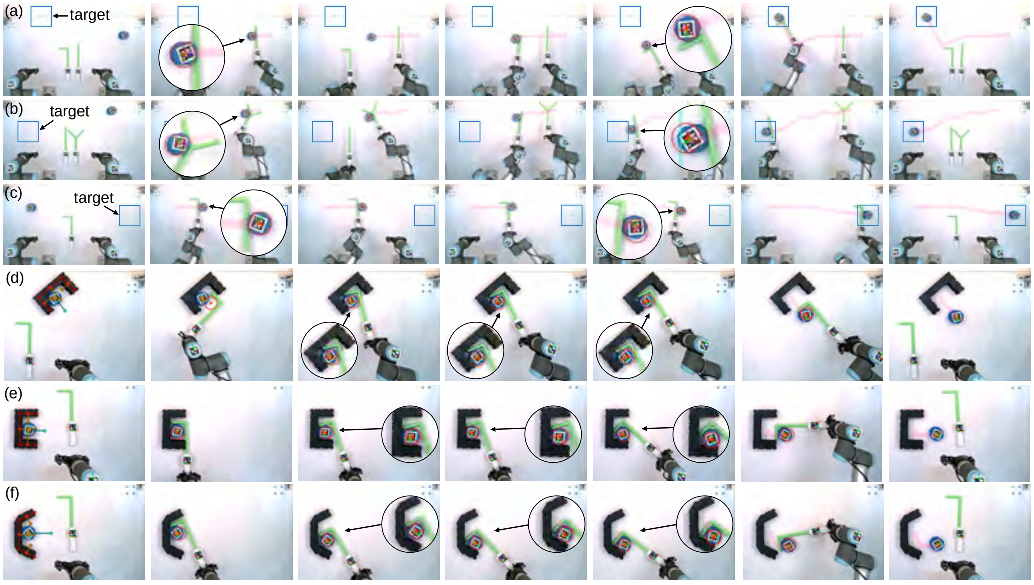


Figure 5.8: Long-horizon task: moving the block from (a) far right to far left with a hook and a stick, (b) far top right to far left with a stick and a Y-shaped tool, (c) far left to far right with a hook; and (d)–(f) exit from a confined area with a stepping controller. The block trajectory is reflected in pink and the target is labelled with a blue square.

manoeuvrability region. We visualise the contact changes between the centre of the highest manoeuvrability region $\mathbf{p}_*$ with the block in Fig. 5.10(a). The error between the $\mathbf{p}^{\text{obj}}$ and the wall exit for each experiment are minimised with time, as shown in Fig. 5.9.

5.3.4 Comparison

We analysed the affordance utilisation and provision for the selected tools. This evaluation involves assessing the frequency of contact between the block and the sides of the tool segments. In the majority of instances, the block interacts with the affordance primarily in the red region, as indicated in Fig. 5.10(b) and aligns closely with our proposed model.

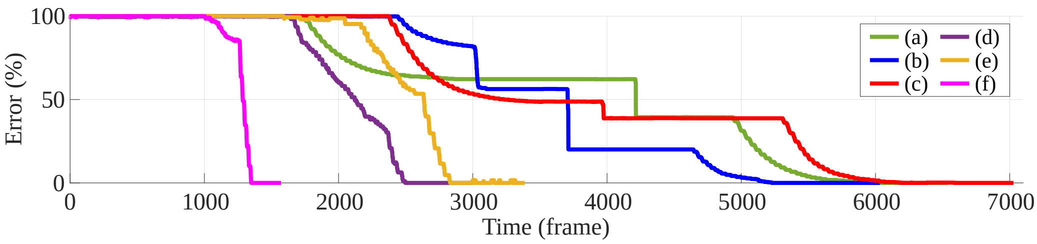


Figure 5.9: Evolution of the minimisation process of the error between the current object position and the target for the tasks shown in Fig. 5.8.

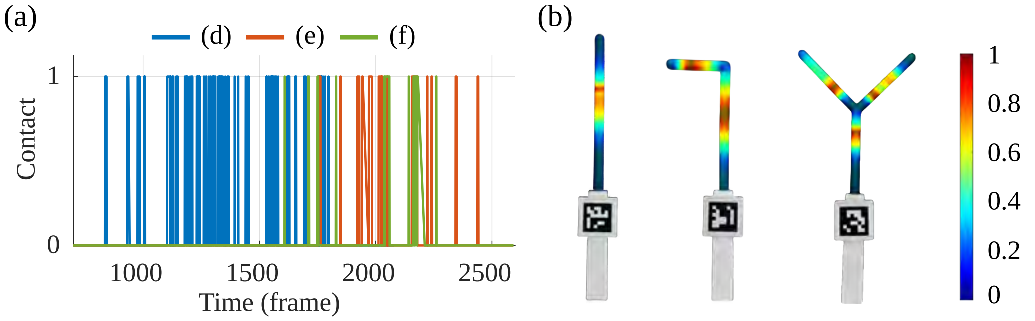


Figure 5.10: (a) Stepping movement evolution of the change in contact between the block and the highest manoeuvrability point for the tasks shown in Fig. 5.8(d)–(f). 1 refers to in-contact and 0 refers to no contact. (b) Contact frequency of a segment side: regions depicted in deeper red indicate higher contact frequency with the block and a higher occurrence of affordance provision.

We have compared our system with other state-of-the-art methods. In terms of the task decomposition with LLM, we compare ours with zero-shot, few-shot learning [119], [120], and a smaller dataset. The results of root mean square error (RMSE) and mean absolute error (MAE) are shown in Table 5.1 where FT states for fine-tunning, SRST states for a single-arm robot with a single tool, Dual refers to dual arms collaboration with two tools, and Sharing refers to tool-sharing collaboration. We observe that prompting (zero-shot and few-shot learning) is relatively unreliable, especially in long-horizon tasks. This may caused by insufficient manipulation examples given in the prompt. A smaller dataset with GPT 3.5 gen-

Table 5.1: Success Rate Comparison for task decomposition

Methods	SRST	Dual	Sharing	TOME	Overall(%)
Zero-shot learning	2/10	1/10	1/10	5/10	22.5%
Few-shot learning	3/10	1/10	2/10	7/10	32.5%
FT (1000 data)	7/10	6/10	7/10	9/10	72.5%
Ours	10/10	9/10	9/10	10/10	95.0%

Table 5.2: Accuracy comparison of manoeuvrability analysis

Methods	RMSE	MAE	Overall
Total variation regularisation [121]	102.4	115.9	109.2
Keypoints-based [122]	31.7	31.5	31.6
Ours	28.6	29.2	28.9

erates an acceptable result, yet, it occasionally provides unnecessary/infeasible steps in long-horizon tasks. In general, all methods demonstrate relatively positive outcomes in TOME, potentially attributed to the task's simplicity: extracting the block from the constrained environment rather than aiming for a specific destination. In summary, employing a larger dataset with GPT 3.5 yields enhanced task decomposition performance, leading to more precise results.

We assess the tool analysis method by identifying the highest manoeuvrability point across 30 tool images, with results outlined in Table 5.2. While the total variation regularisation approach [121] is suitable for the stick case, the results are not satisfactory. The keypoints-inspired approach [122] yields comparable results to ours; however, its accuracy diminishes with increasingly complex shapes. Overall, our approach achieves a better performance in terms of manoeuvrability computation.

5.4 Discussion

In this chapter, we focus on enhancing the manipulation capability of a dual-arm robot with tool usage. In more specific, we present a new manoeuvrability-driven approach for tool-object manipulation. The LLM is integrated for task decomposition, generating collaborative motion sequences for a dual-arm robot system. A compact geometrical-based affordance model for describing the potential functionality and computing the highest manoeuvrability region of a tool is developed. A non-prehensile motion controller and a stepping manipulation model are derived for TOM and incremental movements in a constrained area. Experimental results are reported and analysed for the proposed methodology validation. We illustrate the performance of the proposed methods in the accompanying video https://vimeo.com/917120431.

Our method introduces a new affordance and manoeuvrability paradigm for tool-based object manipulation. To obtain a better performance, we split the model into task decomposition and mathematical motion models. However, the logical fault in the LLM's response may be unseen and thus lead to inappropriate motion. In our experiments, there are a few times that the LLM presents infeasible plans. Moreover, the current affordance model presents promising results with simple geometrical shapes. Dynamics shapes like deformable objects may be complicated to perform accurate modelling. In terms of manoeuvrability, it may be complicated to compute an accurate result for scenes with unstable illumination, low contrast in images, large height differences in objects (tools and the block), etc. We simplified these cases using ArUco code for real-time object tracking in the experiments.

5.4.1 Project Extension

Based on the proposed model, we would like to extend our method to handle deformable objects and/or environments in the coming research study. Specifically, we aim to build upon the current methodology and develop an enhanced version that allows the robot to use tools to manipulate multiple objects simultaneously. The goal is to increase the number of manipulable objects significantly, potentially targeting a group of objects or even particles.

In terms of the LLM aspect, we would like to apply vision-language models to determine the appropriate task and control actions for tool-based manipulation of deformable particles. By leveraging the language understanding and multimodal reasoning capabilities of LLMs, we hope to enable more flexible and adaptive control of the tool-particle interaction. This could involve utilizing the LLM to interpret the current state of the particle system, reason about the desired manipulation objectives, and generate the corresponding control commands for the robot's tool.

Furthermore, we envision incorporating learning-based techniques within the LLM framework to gradually refine the tool control policies through interaction with the deformable particle environment. The synergy between the high-level task reasoning of the LLM and the low-level control of the tool movements could unlock new possibilities for dexterous and versatile particle manipulation.

Chapter 6

Conclusion and Future Works

6.1 Conclusion

In this thesis, we have focused on utilizing capability in various collaboration type, from assistance to enhancement. We have presented three groundbreaking research studies that demonstrate how robotic systems can effectively assist and collaborate among robots, from providing targeted support for specific tasks to autonomously completing complex jobs through collaborative manipulation. Our cases of study spans a diverse range of collaboration types, including argumented perception assistance bot, heterogeneous multi-robot collaboration, and robot collaborative tool-object manipulation.

To begin, we developed a novel welding training platform that leverages RGB-D and HDR cameras to capture the real-time welding process, providing valuable guidance to learners and quantifying their performance. This system assists trainees with argumented perception capability, offering personalized feedback and real-time guidance, reducing the problem of insufficient human trainer and one-to-one comprehensive learning.

Next, we expanded the capabilities of robotic systems by introducing a capability-based task allocation framework. This framework enables the dynamic sharing and utilization of individual robot capabilities, whether through active or passive distributed approach with the proposed ontological knowledge graph. This framework has proven to be highly effective in orchestrating heterogeneous multi-robot collaboration, resource allocation, and task completion.

Finally, we pushed the boundaries of robotic dexterity and enhanced the robot's initial capability by integrating large language models and tool affordance modeling for non-prehensile tool-object manipulation. By leveraging the semantic understanding derived from language models and the maneuverability-driven control of tool-object interactions, our dual-arm robotic platform can now execute intricate manipulation tasks that rival human-level capabilities, even in confined environments.

Throughout these research endeavors, we have implemented the proposed methodologies and rigorously analyzed their feasibility, efficiency, effectiveness, and performance through real-world experimentation. The originality of our work lies in the novel approach to assist humans in real-world welding training, the innovative modeling of heterogeneous multi-robot collaboration, and the advancements in non-prehensile tool manipulation, covering different capability usage feasibility.

In summary, this thesis focuses utilizing capability in various formats, from targeted assistance to enhancement in manipulation tasks. The insights and technical contributions contained herein pave the way for further advancements in the field of interactive and intelligent robotics, unlocking new possibilities for human-machine symbiosis.

6.2 Future Works

As the workplace continues to evolve at a rapid pace, there is a pressing need to develop a comprehensive robotic worker system capable of efficiently conducting a diverse array of tasks. A key focus of our future research will be to develop solutions that enable robots to closely mimic our working behaviors through a seamless integration of assistance, collaboration, and dexterous tool manipulation.

For instance, we aim to deepen our understanding of human welding practices and leverage this knowledge to enhance our robot's ability to predict the welder's next movements and give guidance in advance. To do so, we can apply learning approach (e.g. imitation learning or learn from human/video demonstration) to first teach the robot how to weld with a torch. Then, the model based on its learning experience to reason the 3D geometrical-mechanical relation of the environment to provide accurate 3D motion assistance to human learner. This level of behavioral understanding will be crucial in fostering a truly symbiotic relationship between human experts and their robotic assistants.

Furthermore, we intend to explore a broader range of collaborative strategies within heterogeneous multi-robot systems. This will ensure our solutions can be tailored to meet the specific demands of various work environments. Specifically, we would like to apply such heterogeneous multi-robot systems to real-world scenarios like stocking and rescue missions. In these cases, the robots can utilize and share their specialized capabilities based on the task and environmental requirements. Beyond ground-level collaborations, we would also like to explore airlevel collaborations. This could involve integrating non-mobile robots with aerial platforms, allowing the robots to collectively overcome hardware limitations and expand their capabilities. By assembling aerial and ground-based robots, the combined system can tackle more challenging missions that exceed the initial capabilities of individual units. By expanding the repertoire of collaborative strategies across heterogeneous robots, we can unlock new avenues for robotic teams to tackle increasingly complex challenges in a coordinated and efficient manner. This multi-modal approach to collaboration will enable robotic systems to better adapt to diverse work environments and tackle a wider range of tasks.

Moving forward, we aim to expand the capabilities of collaborative manipulation, focusing on the control of object and material shapes. Specifically, we aim to develop dual-arm robotic systems capable of precisely manipulating the shape of particles and other fine-grained materials through the dexterous use of tools in a confined situation with the use of Fourier series and model predictive control. This line of research holds great promise in areas such as additive manufacturing, material processing, and even medical applications, where the ability to delicately control material properties could yield transformative advancements.

By consolidating the research insights gained from the three case studies, our next primary research focus will be on developing a collaborative heterogeneous multi-robot framework that integrates tool manipulation capabilities for shape control. In this upcoming study, we plan to synergistically apply the capability-based collaboration model we have previously established, combined with a vision-language model for task decomposition. The tool affordance and usage will be derived from the methodologies proposed in our earlier work. To address the shape control of objects, such as particles, we will leverage Fourier series to represent the macroscopic shape of the particle ensemble. Additionally, we will incorporate model predictive control techniques to enable obstacle avoidance during the non-prehensile tool manipulation tasks.

Appendix A

Educational Platforms

This appendix details the educational systems developed as part of the research conducted during the doctoral study.

In addition to the primary focus on interaction-oriented methodologies, this thesis also explores new teaching and learning approaches to better understand effective knowledge delivery and adoption. To validate the proposed hypotheses, specialized educational platforms were developed and implemented.

A.1 Background

Mastering robotic systems requires a diverse set of knowledge and skills from different disciplines such as electronics, mechanics, design, programming, and engineering. Effectively passing on this wealth of wisdom and knowledge to the next generation is often a crucial challenge that demands innovative teaching approaches.

Traditional lecturer-centered teaching methods often prove inadequate for learning systems and control theory. These approaches tend to encourage surface learning, where students memorize content merely to pass examinations, without truly grasping the underlying principles [123]. This situation is further exacerbated by the general lack of interest in analytical problems observed among many undergraduate students. In the current teaching landscape, it is common to encounter the ”Susan and Robert” phenomenon, where ”Susan” refers to academically-inclined and motivated students, while ”Robert” represents those who are less interested and seek only to obtain a qualification [124]. The passive, lecturer-centered approach that works well for ”Susan”-type students may not effectively engage ”Robert” due to the lack of active student involvement.

To bridge the gap between ”Susan” and ”Robert,” a modern, engaging flipped classroom strategy should be incorporated. Inspired by the work of [125], a participation-based approach that focuses on constructing knowledge through active student engagement in activities has been developed. During my Ph.D. studies, I was involved as a teaching assistant in the development of flipped classroom materials and the implementation of participation-based teaching and learning platforms.

A.2 Methodology

Many of the concepts introduced in lectures can be considered intangible assets, making them challenging for students to fully comprehend without practical experience. However, some of these theoretical concepts cannot be easily demonstrated or allow for hands-on trial-and-error practice. To enrich students’ understanding of the taught theories and encourage them to explore potential applications, a simple, safe, and cost-effective teaching and learning platform is in high demand.

To address this need, two educational platforms were developed during my PhD studies: (1) a LEGO-based system and (2) a physics-based control game. These platforms aim to provide students with tangible, interactive experiences to deepen their grasp of the underlying principles and stimulate their interest in exploring further applications.

A.2.1 LEGO-based Mass-Spring-Damper System

To strengthen the theory taught in lectures with hands-on application, we decided to leverage the well-known building toy: LEGO.

LEGO has been enjoyed by children, teenagers, and even adults for many years, and has proven its widespread appeal, positive user experience, durability, high fault tolerance, and value in inspiring creativity and imaginative structures. Importantly, LEGO also has the ability to simplify complex concepts into straightforward, modular components, making it an ideal platform for the simplified, yet practical, control applications we aim to explore.

To develop a tangible learning platform centered around mass-spring-damper systems, we utilized LEGO components to construct a simple 2-DOF system, as illustrated in Fig. A.1. The cart design was engineered to be user-friendly, allowing students to easily adjust the mass or weight by adding loads onto the cart. Additionally, the number of degrees of freedom (DOF), represented by the number of carts, can be easily modified by adding or subtracting carts.

To guide the lateral motion of the cart, a single-track railway-like trail was

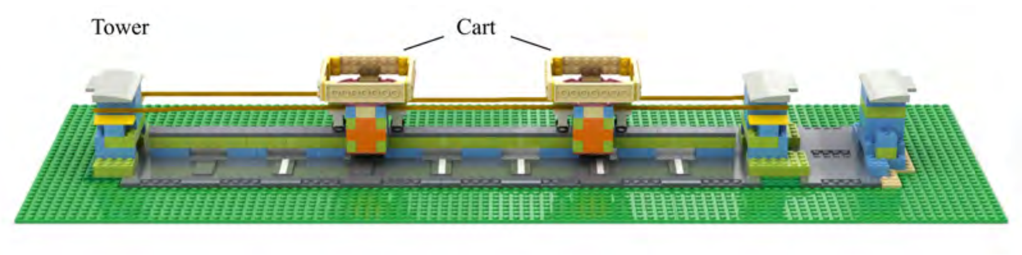


Figure A.1: The LEGO-based mass-spring-damper system allows students to construct and experiment with a simplified representation of this fundamental control system. By building their own carts, students can adjust parameters such as the mass and number of carts, the stiffness of the spring using different rubber band lengths, and the damping effect by incorporating creative structural components.

implemented. The vertical motion is controlled by the combined effect of gravity and the weight of the cart plus any added loads. To ensure a smooth linear lateral movement along the trail, a pair of steering wheels were mounted upside-down to act as low-friction wheels.

To simulate the spring effect in the mass-spring-damper system, we leveraged a common everyday material – rubber bands. The cart and the tower are connected by a rubber band, allowing students to control the system stiffness by adjusting the rubber band's properties, such as thickness, length, and the pulling distance between the cart and tower.

The damping effect is introduced through friction. While the friction between the cart's wheels and the trail can be considered negligible, we added an external component onto the cart with a rubber band wrapped around a block. This setup creates significant friction, acting as a damper to reduce the vibration frequency and cause the cart to eventually come to a stop.

The vibration frequency of the system is measured using a proximity sensor connected to an Arduino Uno, as shown in Fig. A.2. The sensor is mounted on

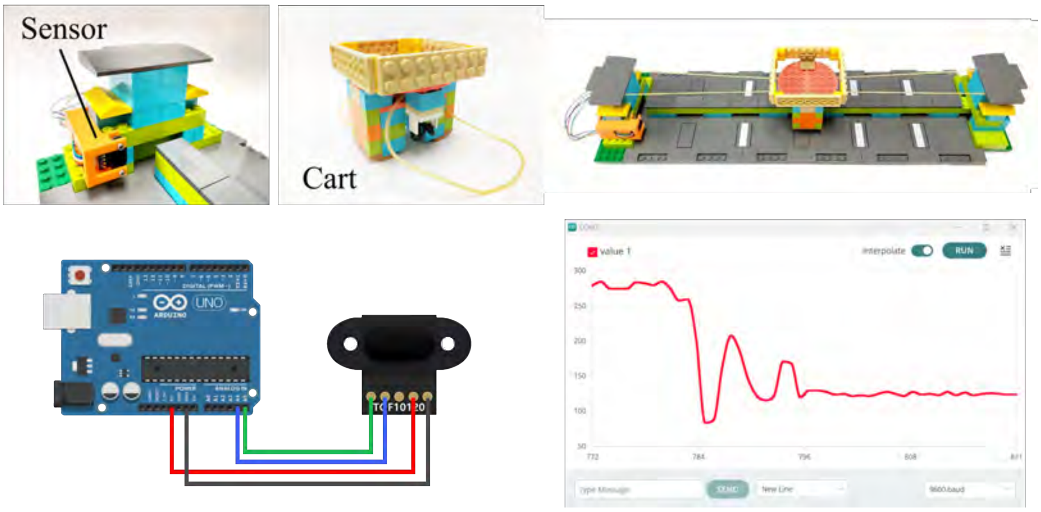


Figure A.2: A proximity sensor is mounted on the tower to record the changing distance between the tower and the cart. The sensor is connected to an Arduino micro-controller, which allows the vibration frequency of the mass-spring-damper system to be visualized over time. The line graph demonstrates how the vibration frequency is reduced as a damper is added to the cart, illustrating the system's dynamic behavior.

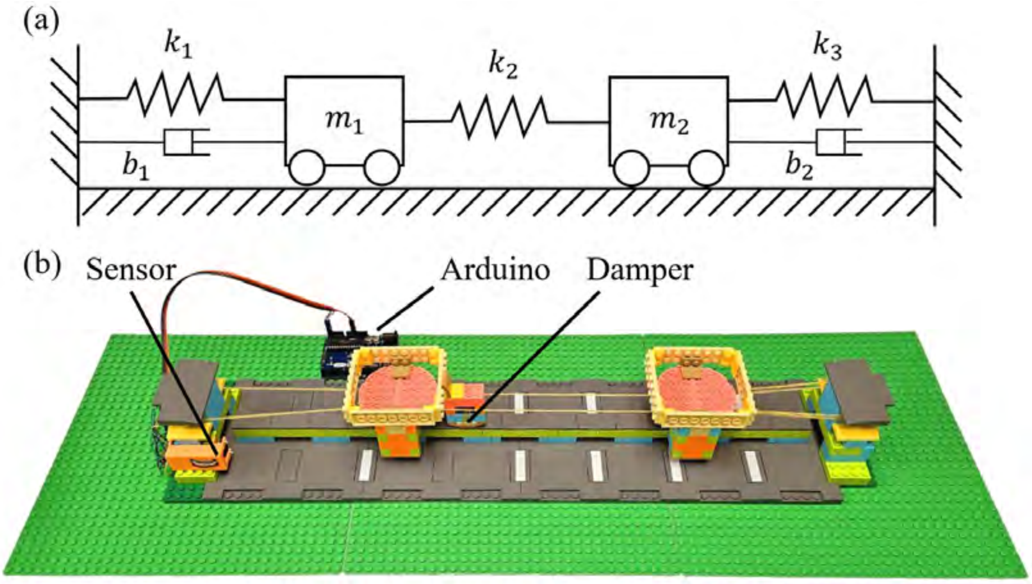


Figure A.3: Traditional mass-spring-damper system concepts, often presented only theoretically in textbooks, can be visually realized using LEGO components. This hands-on representation allows students to tangibly explore and experiment with the underlying principles behind these fundamental control systems.

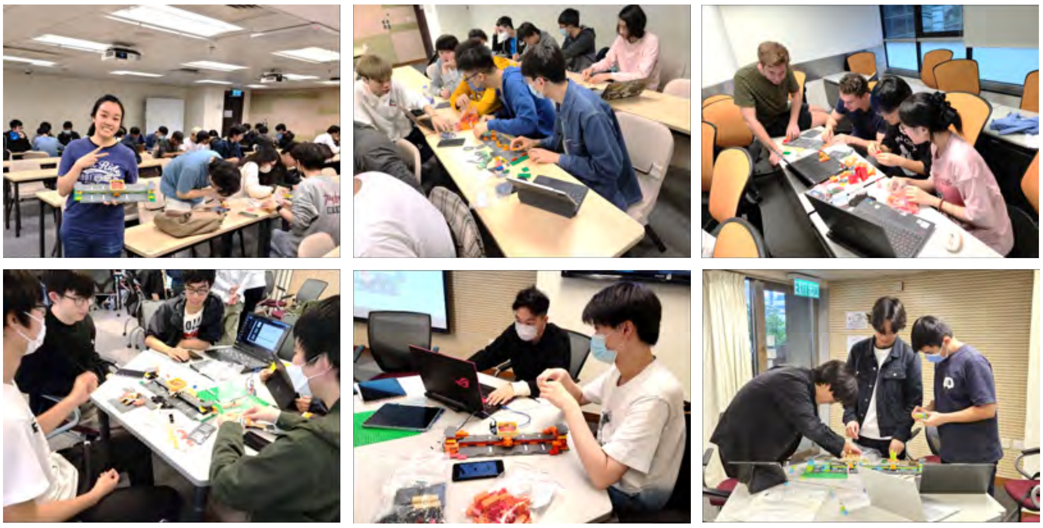


Figure A.4: The LEGO-based mass-spring-damper system has been successfully integrated into two mechanical engineering courses. In the first photo, the instructor (which is me), wearing a blue t-shirt, is demonstrating the LEGO system to students during a lab session. The subsequent images show students actively engaged in constructing and experimenting with the hands-on learning platform.

the side of the tower, facing the cart, to record the distance changes between the cart and the tower. The amplitude of the system's vibration can then be analyzed by visualizing the data plot in the Arduino IDE.

This LEGO-based mass-spring-damper learning platform has been successfully deployed in two mechanical engineering courses: ME31002 Linear Systems and Control, and ME31003 System Dynamics, as a practical lab session, as shown in Fig. A.4.

During the lab, students are first provided with detailed instructions to build a 1-DOF mass-spring system using the LEGO components and electronics. Once they have familiarized themselves with the hardware, they are then tasked with designing and building a 2-DOF mass-spring-damper system as an assignment. This challenge requires students to use their creativity and the remaining LEGO parts to develop an effective damper mechanism. After the lab session, the students are required to submit a report and a video, in which they explain their design rationale, demonstrate their understanding of the mass-spring-damper system concept, and showcase the application of different stiffness and damping settings. More details can be found in https://romi-lab.github.io/msds/.

This active, hands-on learning approach has proven effective in motivating not only the academically-inclined ”Susan” students but also the less engaged ”Robert” students to focus on and learn the intangible control theory concepts through the tangible trial-and-error experience with the LEGO-based mass-spring-damper system.

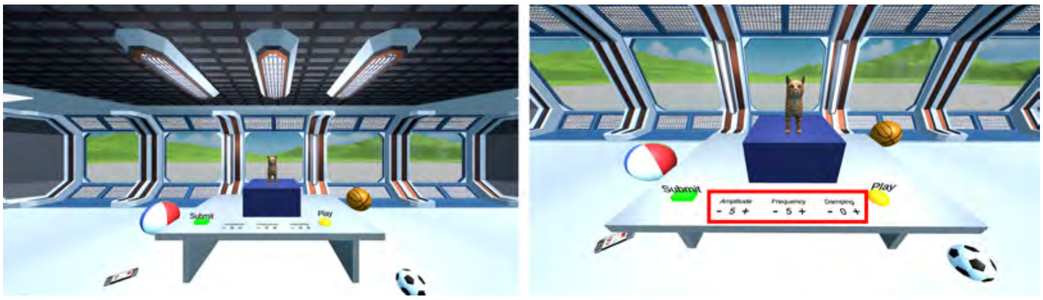


Figure A.5: Users can control the amplitude, frequency, and damping effects by adjusting the parameters of the mass-spring-damper equation as shown in the red box.

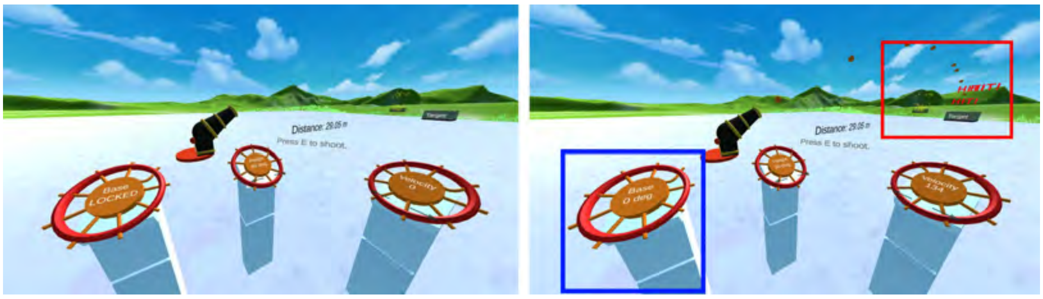


Figure A.6: Users can control the shooting angles based on the distance between the target and the cannon by rotating the valves as shown in the blue box. Real-time feedback is given if the target is hit.

A.2.2 Physics and Control System

While the LEGO-based mass-spring-damper system provides students with valuable hands-on experience in constructing and experimenting with the control theory concepts taught in lectures, it lacks the ability to provide precise quantitative control over the system parameters, such as the actual mass, spring stiffness, and damping.

To enhance the teaching and learning experience, a physics-based control system simulation game was developed. This virtual laboratory environment empowers students to test their ideas with quantitative control over the system properties

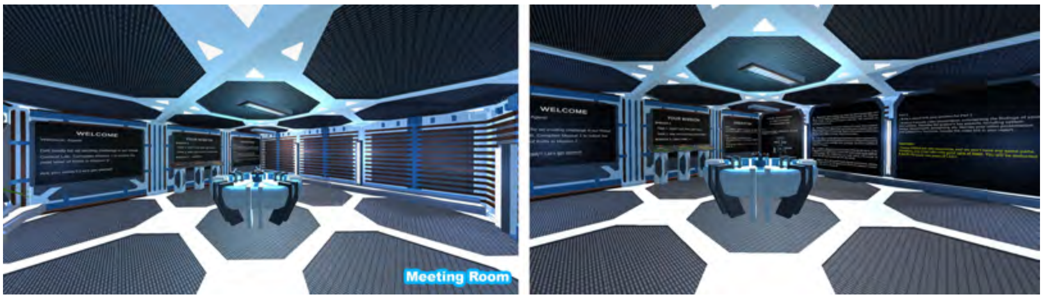
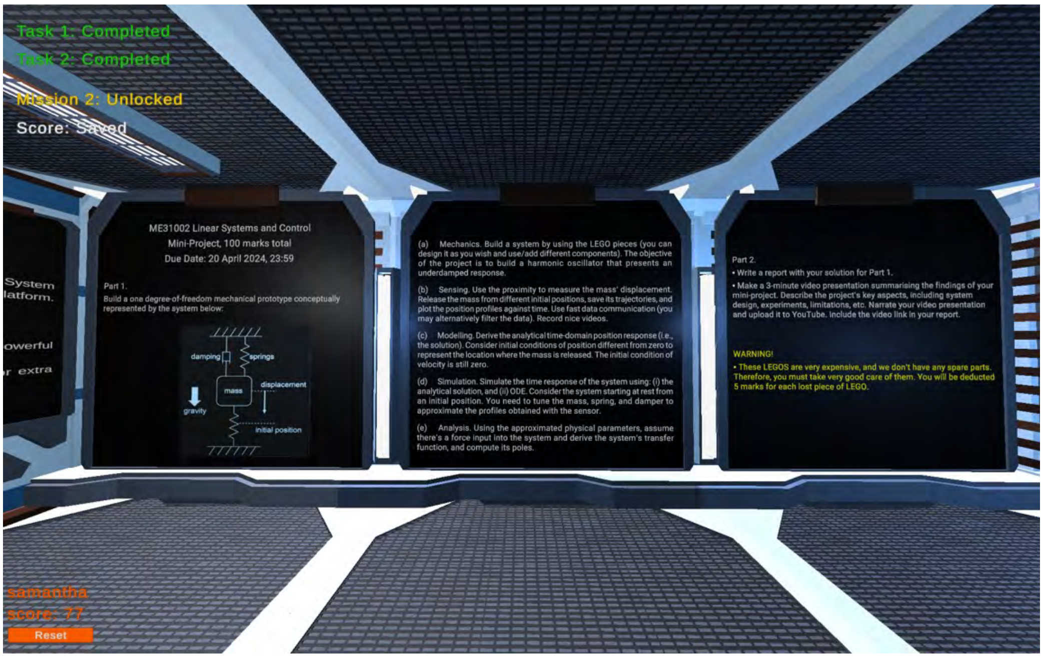


Figure A.7: Task description is shown on the blackboard. Left: before the completion of tasks with Mission 2 locked. Right: after the completion of Mission 1 with Mission 2 unlocked.



Task 1: Completed

Task 2: Completed

Mission 2: Unlocked

Score: Saved

ME31002 Linear Systems and Control
Mini-Project, 100 marks total
Due Date: 20 April 2024, 23:59

Part 1.

Build a one degree-of-freedom mechanical prototype conceptually represented by the system below:

(a) Mechanics. Build a system by using the LEGO pieces (you can design it as you wish and use/add different components). The objective of the project is to build a harmonic oscillator that presents an underdamped response.

(b) Sensing. Use the proximity to measure the mass' displacement. Release the mass from different initial positions, save its trajectories, and plot the position profiles against time. Use fast data communication (you may alternatively filter the data). Record nice videos.

(c) Modelling. Derive the analytical time-domain position response (i.e., the solution). Consider initial conditions of position different from zero to represent the location where the mass is released. The initial condition of velocity is still zero.

(d) Simulation. Simulate the time response of the system using: (i) the analytical solution, and (ii) ODE. Consider the system starting at rest from an initial position. You need to tune the mass, spring, and damper to approximate the profiles obtained with the sensor.

(e) Analysis. Using the approximated physical parameters, assume there's a force input into the system and derive the system's transfer function, and compute its poles.

Part 2.

- Write a report with your solution for Part 1.
- Make a 3-minute video presentation summarising the findings of your mini-project. Describe the project's key aspects, including system design, experiments, limitations, etc. Narrate your video presentation and upload it to YouTube. Include the video link in your report.

WARNING!

- These LEGOS are very expensive, and we don't have any spare parts. Therefore, you must take very good care of them. You will be deducted 5 marks for each lost piece of LEGO.

samantha
score: 77
Reset

Figure A.8: The description of Mission 2 is shown on the blackboard, showing the requirement and guidelines for the LEGO-based mass-spring-damper system challenge.

and simulate real-world scenarios. Leveraging the strengths of the LEGO-based approach, the simulation game includes two main missions:

- Mission 1:
    1. Task 1: Cat Vibration System (see Fig.A.5)
    2. Task 2: Cannon Shooting System (see Fig.A.6)
- Mission 2:
    1. Hands-on Practice with LEGO-based Mass-Spring-Damper System
    2. Results Reporting with Video Explanation

In these missions, students are encouraged to adjust the system parameters and observe the immediate effects, allowing them to build an intuitive understanding of the underlying principles.

In the “Cat Vibration System” mission, the blue block acts as a vibrator, moving horizontally. By default, the cat is unable to remain standing on the vibrating platform. The goal of this task is for the student to adjust the vibration parameters, as shown on the desk in Fig. A.5, to make the cat stay on the vibrator for more than 30 seconds. To encourage further exploration, the system includes some ”Easter eggs,” where students can experiment with the vibration effects on objects of different masses and friction levels to earn bonus points. For example, placing a banana or a basketball on the vibrator and observing the results can earn an additional 10 points.

The “Cannon Shooting System” mission features two targets and three valves that allow students to adjust the cannon’s angle and shooting velocity, as depicted in Fig. A.6. The distance between the cannon and the target is provided to assist with aiming. When the target is hit, a red “Hit!” sign is immediately displayed next to it. Upon successfully hitting both targets, Mission 2 is unlocked, and the mission description is shown on the blackboard inside the meeting room, as illustrated in Fig. A.7 and A.8.

The missions are designed to be unlocked sequentially, with students required to complete all virtual tasks before gaining access to the physical practice tasks. The user’s performance is displayed in the top-left corner, allowing both the instructor and the students to track their progress. Additionally, a “Reset” button is provided to encourage students to replay the game and apply the learnt concepts to achieve a higher score. More details can be found in https://romi-lab.github.io/is-everything-under-control/.

The comprehensive learning experience is achieved by seamlessly combining the virtual physics and control system simulation with the hands-on LEGO-based experimentation. This blended approach not only reinforces the theoretical concepts taught in lectures but also encourages both high-performing “Susan” students and less engaged “Robert” students to actively participate in applying the newly learnt principles through practical, tangible implementations. By alternating between the virtual simulations and the physical LEGO system, students are able to develop a deeper, more intuitive understanding of the mass-spring-damper system and the associated control theory.

List of Publications

1. **H.-Y. Lee**, P. Zhou, A. Duan, W. Ma, C. Yang, and D. Navarro-Alarcon, “Non-Prehensile Tool-Object Manipulation by Integrating LLM-Based Planning and Manoeuvrability-Driven Controls”. (preprint)

2. **H.-Y. Lee**, P. Zhou, B. Zhang, L. Qiu, and D. Navarro-Alarcon, “A Distributed Dynamic Framework to Allocate Collaborative Tasks Based on Capability Matching in Heterogeneous Multi-Robot Systems”, *IEEE Transactions on Cognitive and Developmental Systems (T-CDS)*, 2023.

3. **H.-Y. Lee**, P. Zhou, A. Duan, V. Wu, and D. Navarro-Alarcon, “A Multi-Sensor Interface to Improve the Learning Experience in Arc Welding Training Tasks”, *IEEE Transactions on Human-Machine Systems (T-HMS)*, 2023.

4. J. Qi, P. Zhou, P. Zheng, C. Li, L. Lu, Z. Zhang, Y. Ding, **H.-Y. Lee**, D. Navarro-Alarcon, and J. Pan, “A Coarse-to-Fine Framework for Robotic Fabric Alignment System: Integrating Visual Servoing and Admittance Control”. (preprint)

5. W. Ma, A. Duan, **H.-Y. Lee**, P. Zheng, and D. Navarro-Alarcon, “Human-Aware Reactive Task Planning of Sequential Robotic Manipulation Tasks”,

Proc. IEEE Transactions on Industrial Informatics (T-II), under review, 2024

6. H. Chen, S. Huo, M. Muddassir, **H.-Y. Lee**, A. Duan, P. Zheng, H. Pan, D. Navarro-Alarcon, "PSO-Based Optimal Coverage Path Planning for Surface Defect Inspection of 3C Components with a Robotic Line Scanner". (preprint)

7. P. Zhou, P. Zheng, J. Qi, C. Li, **H.-Y. Lee**, A. Duan, L. Lu, Z. Li, L. Hu, D. Navarro-Alarcon, "Reactive Human-Robot Collaborative Manipulation of Deformable Linear Objects Using a New Topological Latent Control Model", *Robotics and Computer-Integrated Manufacturing*, 88: 102727, 2024.

8. A. Duan, S. Huo, **H.-Y. Lee**, P. Zhou, J. G. Romero, C. Yang and D. Navarro-Alarcon, "Robust Grasping by Bimanual Robots with Stable Parametrization-based Contact Servoing", *IEEE Transactions on Mechatronics (T-MECH)*, 2024.

9. L. Labazanova, S. Peng, L. Qiu, **H.-Y. Lee**, T. Nanayakkara, and D. Navarro-Alarcon, "Self-Reconfigurable Soft-Rigid Mobile Agent with Variable Stiffness and Adaptive Morphology", *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 3, pp. 1643-1650, 2023.

Bibliography

[1] P. K. Ghosh, *Pulse current gas metal arc welding*. Springer, 2017.

[2] M. I. Khan, *Welding science and technology*. New Age International, 2007.

[3] C. Latella, Y. Tirupachuri, L. Tagliapietra, *et al.*, "Analysis of human whole-body joint torques during overhead work with a passive exoskeleton," *IEEE Transactions on Human-Machine Systems*, 2021.

[4] Y. Liu, W. Zhang, and Y. M. Zhang, "A tutorial on learning human welder's behavior: Sensing, modeling, and control," *Journal of Manufacturing processes*, vol. 16, no. 1, pp. 123–136, 2014.

[5] S.-B. Asplund and N. Kilbrink, "Lessons from the welding booth: Theories in practice in vocational education," *Empirical Research in Vocational Education and Training*, vol. 12, no. 1, pp. 1–23, 2020.

[6] K. Weman, *Welding processes handbook*. Elsevier, 2011.

[7] J. M. Antonini, "Health effects of welding," *Critical reviews in toxicology*, vol. 33, no. 1, pp. 61–103, 2003.

[8] A. D. Althouse, C. H. Turnquist, W. A. Bowditch, K. E. Bowditch, and M. A. Bowditch, *Modern welding*. Goodheart-Wilcox Publisher, 2004.

[9] R. Peng, D. Navarro-Alarcon, V. Wu, and W. Yang, “A point cloud-based method for automatic groove detection and trajectory generation of robotic arc welding tasks,” *2020 17th International Conference on Ubiquitous Robots (UR)*, pp. 380–386, 2020.

[10] P. Zhou, R. Peng, M. Xu, V. Wu, and D. Navarro-Alarcon, “Path planning with automatic seam extraction over point cloud models for robotic arc welding,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5002–5009, 2021.

[11] C.-c. Chung, C.-C. Tung, and S.-J. Lou, “Research on optimization of vr welding course development with anp and satisfaction evaluation,” *Electronics*, vol. 9, no. 10, p. 1673, 2020.

[12] C.-Y. Huang, S.-J. Lou, Y.-M. Cheng, and C.-C. Chung, “Research on teaching a welding implementation course assisted by sustainable virtual reality technology,” *Sustainability*, vol. 12, no. 23, p. 10 044, 2020.

[13] D.-S. Jo, Y.-W. Kim, U.-Y. Yang, G.-A. Lee, J.-S. Choi, and K.-H. Kim, “Virtual reality based welding training simulator,” in *Proceedings of the KWS Conference*, The Korean Welding and Joining Society, 2010, pp. 49–49.

[14] M. I. M. Isham, H. N. H. Haron, F. bin Mohamed, C. V. Siang, M. K. Mokhtar, and A. S. binti Azizo, “Mobile vr and marker tracking method applied in virtual welding simulation kit for welding training,” in *2020 6th International Conference on Interactive Digital Media (ICIDM)*, IEEE, 2020, pp. 1–5.

[15] U. Yang, G. A. Lee, Y. Kim, D. Jo, J. Choi, and K.-H. Kim, "Virtual reality based welding training simulator with 3d multimodal interaction," in *2010 International Conference on Cyberworlds*, IEEE, 2010, pp. 150–154.

[16] T. L. Chambers, A. Aglawe, D. Reiners, *et al.*, "Real-time simulation for a virtual reality-based mig welding training system," *Virtual Reality*, vol. 16, no. 1, pp. 45–55, 2012.

[17] D. Mavrikios, V. Karabatsou, D. Fragos, and G. Chryssolouris, "A prototype virtual reality-based demonstrator for immersive and interactive simulation of welding processes," *International Journal of Computer Integrated Manufacturing*, vol. 19, no. 03, pp. 294–300, 2006.

[18] Y. Fangming, "Real-time construction of 3d welding torch in virtual space for welding training simulator," *International Journal of Engineering and Manufacturing*, vol. 9, no. 5, pp. 34–45, 2019.

[19] K. Kobayashi, S. Ishigame, and H. Kato, "Skill training system of manual arc welding," in *Entertainment Computing*, Springer, 2003, pp. 389–396.

[20] S. Doolani, C. Wessels, V. Kanal, *et al.*, "A review of extended reality (xr) technologies for manufacturing training," *Technologies*, vol. 8, no. 4, p. 77, 2020.

[21] R. S. Vergel, P. M. Tena, S. C. Yrurzum, and C. Cruz-Neira, "A comparative evaluation of a virtual reality table and a hololens-based augmented reality system for anatomy training," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 4, pp. 337–348, 2020.

[22] R. Agrawal and J. S. Pillai, "Augmented reality application in vocational education: A case of welding training," in *Companion Proceedings of the 2020 Conference on Interactive Surfaces and Spaces*, 2020, pp. 23–27.

[23] A. W. Society, *Aws a3. 0m/a3. 0: 2010: Standard welding terms and definition*, 2010.

[24] L. Ding and A. Goshtasby, "On the canny edge detector," *Pattern Recognition*, vol. 34, no. 3, pp. 721–725, 2001.

[25] X. Xiong, G. Jin, Q. Xu, H. Zhang, and J. Xu, "Robust line detection of synthetic aperture radar images based on vector radon transformation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 5310–5320, 2019.

[26] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM*, vol. 18, no. 9, pp. 509–517, 1975.

[27] Y. Fang, S. Ko, and G.-S. Jo, "Robust visual tracking based on global-and-local search with confidence reliability estimation," *Neurocomputing*, vol. 367, pp. 273–286, 2019.

[28] R. Yasarla and V. M. Patel, "Confidence measure guided single image de-raining," *IEEE Transactions on Image Processing*, vol. 29, pp. 4544–4555, 2020.

[29] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2547–2554.

[30] B. Polster, *The shoelace book: a mathematical guide to the best (and worst) ways to lace your shoes*. American Mathematical Soc., 2006.

[31] B. Millidge, A. Tschantz, A. Seth, and C. Buckley, "Neural kalman filtering," *arXiv preprint arXiv:2102.10021*, 2021.

[32] M. J. Anzanello and F. S. Fogliatto, "Learning curve models and applications: Literature review and research directions," *International Journal of Industrial Ergonomics*, vol. 41, no. 5, pp. 573–583, 2011.

[33] G. Fioretti, "The organizational learning curve," *European Journal of Operational Research*, vol. 177, no. 3, pp. 1375–1384, 2007.

[34] A. Newell and P. S. Rosenbloom, "Mechanisms of skill acquisition and the law of practice," *Cognitive skills and their acquisition*, vol. 1, no. 1981, pp. 1–55, 1981.

[35] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[36] F. Wang, J. Cheng, W. Liu, and H. Liu, "Additive margin softmax for face verification," *IEEE Signal Processing Letters*, vol. 25, no. 7, pp. 926–930, 2018.

[37] E. Montijano and C. Sagüés, *Distributed consensus with visual perception in multi-robot systems*. Springer, 2015.

[38] Y. Rizk, M. Awad, and E. W. Tunstel, "Decision making in multiagent systems: A survey," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 514–529, 2018.

[39] D. Tarapore, D. Floreano, and L. Keller, “Task-dependent influence of genetic architecture and mating frequency on division of labour in social insect societies,” *Behavioral Ecology and Sociobiology*, vol. 64, no. 4, pp. 675–684, 2010.

[40] C. Liu, Q. Lin, H. Kim, and M. Yim, “Smores-ep, a modular robot with parallel self-assembly,” *arXiv preprint arXiv:2104.00800*, 2021.

[41] R. E. Wang, S. A. Wu, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, “Too many cooks: Coordinating multi-agent collaboration through inverse planning,” 2020.

[42] J.-F. Boudet, J. Lintuvuori, C. Lacouture, *et al.*, “From collections of independent, mindless robots to flexible, mobile, and directional superstructures,” *Science Robotics*, vol. 6, no. 56, eabd0272, 2021.

[43] J. Li, J. Wu, J. Li, A. K. Bashir, M. J. Piran, and A. Anjum, “Blockchain-based trust edge knowledge inference of multi-robot systems for collaborative tasks,” *Communications Magazine*, vol. 59, no. 7, pp. 94–100, 2021.

[44] L. Yan, T. Stouraitis, and S. Vijayakumar, “Decentralized ability-aware adaptive control for multi-robot collaborative manipulation,” *Robot. Autom. Lett.*, vol. 6, no. 2, pp. 2311–2318, 2021.

[45] Y. Zhou, J. Xiao, Y. Zhou, and G. Loianno, “Multi-robot collaborative perception with graph neural networks,” *Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2289–2296, 2022.

[46] Y. Huang, Y. Zhang, and H. Xiao, “Multi-robot system task allocation mechanism for smart factory,” in *8th Joint Int. Information Technology and Artificial Intelligence Conf.*, IEEE, 2019, pp. 587–591.

[47] Y. Emam, S. Mayya, G. Notomista, A. Bohannon, and M. Egerstedt, "Adaptive task allocation for heterogeneous multi-robot teams with evolving and unknown robot capabilities," in *Int. Conf. on Robotics and Automation*, IEEE, 2020, pp. 7719–7725.

[48] S. Mayya, D. S. D'antonio, D. Saldaña, and V. Kumar, "Resilient task allocation in heterogeneous multi-robot systems," *Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1327–1334, 2021.

[49] A. Prorok, M. A. Hsieh, and V. Kumar, "The impact of diversity on optimal control policies for heterogeneous robot swarms," *Trans. on Robotics*, vol. 33, no. 2, pp. 346–358, 2017.

[50] M. A. Karimi, V. Alizadehyazdi, H. M. Jaeger, and M. Spenko, "A self-reconfigurable variable-stiffness soft robot based on boundary-constrained modular units," *IEEE Transactions on Robotics*, 2021.

[51] A. L. Christensen, R. O'Grady, and M. Dorigo, "Swarmorph-script: A language for arbitrary morphology generation in self-assembling robots," *Swarm Intelligence*, vol. 2, no. 2, pp. 143–165, 2008.

[52] R. O'Grady, A. L. Christensen, and M. Dorigo, "Autonomous reconfiguration in a self-assembling multi-robot system," in *International Conference on Ant Colony Optimization and Swarm Intelligence*, Springer, 2008, pp. 259–266.

[53] S. Kernbach, O. Scholz, K. Harada, *et al.*, "Multi-robot organisms: State of the art," *arXiv preprint arXiv:1108.5543*, 2011.

[54] Y. Tu, G. Liang, and T. L. Lam, “Freesn: A freeform strut-node structured modular self-reconfigurable robot-design and implementation,” in *2022 Int. Conf. on Robotics and Automation. IEEE*, 2022.

[55] D. Zhao and T. L. Lam, “Snailbot: A continuously dockable modular self-reconfigurable robot using rocker-bogie suspension,” in *2022 Int. Conf. on Robotics and Automation*, IEEE, 2022, pp. 4261–4267.

[56] G. Liang, H. Luo, M. Li, H. Qian, and T. L. Lam, “Freebot: A freeform modular self-reconfigurable robot with arbitrary connection point-design and implementation,” in *Int. Conf. on Intelligent Robots and Systems*, IEEE, 2020, pp. 6506–6513.

[57] K. Macarthur, R. Stranders, S. Ramchurn, and N. Jennings, “A distributed anytime algorithm for dynamic task allocation in multi-agent systems,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, 2011, pp. 701–706.

[58] G. Anders, C. Hinrichs, F. Siefert, P. Behrmann, W. Reif, and M. Sonnenschein, “On the influence of inter-agent variation on multi-agent algorithms solving a dynamic task allocation problem under uncertainty,” in *2012 IEEE Sixth International Conference on Self-Adaptive and Self-Organizing Systems*, IEEE, 2012, pp. 29–38.

[59] Q. Yang, Z. Luo, W. Song, and R. Parasuraman, “Self-reactive planning of multi-robots with dynamic task assignments,” in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, IEEE, 2019, pp. 89–91.

[60] K. Lerman, C. Jones, A. Galstyan, and M. J. Matarić, "Analysis of dynamic task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 25, no. 3, pp. 225–241, 2006.

[61] M. Chen and D. Zhu, "A workload balanced algorithm for task assignment and path planning of inhomogeneous autonomous underwater vehicle system," *IEEE transactions on cognitive and developmental systems*, vol. 11, no. 4, pp. 483–493, 2018.

[62] C. Moulin-Frier, T. Fischer, M. Petit, *et al.*, "Dac-h3: A proactive robot cognitive architecture to acquire and express knowledge about the world and the self," *IEEE Trans. on Cognitive and Developmental Systems*, vol. 10, no. 4, pp. 1005–1022, 2017.

[63] J. Ginés, F. Martín, D. Vargas, F. J. Rodríguez, and V. Matellán, "Social navigation in a cognitive architecture using dynamic proxemic zones," *Sensors*, vol. 19, no. 23, p. 5189, 2019.

[64] J. Akroyd, S. Mosbach, A. Bhave, and M. Kraft, "Universal digital twin-a dynamic knowledge graph," *Data-Centric Engineering*, vol. 2, 2021.

[65] X. Zhou, A. Eibeck, M. Q. Lim, N. B. Krdzavac, and M. Kraft, "An agent composition framework for the j-park simulator-a knowledge graph for the process industry," *Computers & Chemical Engineering*, vol. 130, p. 106 577, 2019.

[66] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks," in *Graph Theoretic Methods in Multiagent Networks*, Princeton University Press, 2010.

[67] J. Bai, L. Cao, S. Mosbach, J. Akroyd, A. A. Lapkin, and M. Kraft, "From platform to knowledge graph: Evolution of laboratory automation," *JACS Au*, vol. 2, no. 2, pp. 292–309, 2022.

[68] B. Smith, "Ontology," in *The furniture of the world*, Brill, 2012, pp. 47–68.

[69] A. Olivares-Alarcos, D. Beßler, A. Khamis, *et al.*, "A review and comparison of ontology-based approaches to robot autonomy," *The Knowledge Engineering Review*, vol. 34, 2019.

[70] C. R. Garrett, R. Chitnis, R. Holladay, *et al.*, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[71] F. L. Ratnieks and C. Anderson, "Task partitioning in insect societies," *Insectes sociaux*, vol. 46, no. 2, pp. 95–108, 1999.

[72] E. A. Sisbot, R. Ros, and R. Alami, "Situation assessment for human-robot interactive object manipulation," in *2011 RO-MAN*, IEEE, 2011, pp. 15–20.

[73] P. Song, W. Zheng, Y. Yu, and S. Ou, "Speech emotion recognition based on robust discriminative sparse regression," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 13, no. 2, pp. 343–353, 2020.

[74] B. Hanin, "Universal function approximation by deep neural nets with bounded width and relu activations," *Mathematics*, vol. 7, no. 10, p. 992, 2019.

[75] M. Mancini, E. Ricci, B. Caputo, and S. R. Bulò, “Adding new tasks to a single network with weight transformations using binary masks,” in *European Conference on Computer Vision*, Springer, 2018, pp. 180–189.

[76] S. Santra and P. P. Acharjya, “A study and analysis on computer network topology for data communication,” *International Journal of Emerging Technology and Advanced Engineering*, vol. 3, no. 1, pp. 522–525, 2013.

[77] N. Seddigh and M. Devetsikiotis, “Studies of tcp’s retransmission timeout mechanism,” in *ICC 2001. IEEE International Conference on Communications. Conference Record (Cat. No. 01CH37240)*, IEEE, vol. 6, 2001, pp. 1834–1840.

[78] M. Lagomarsino, M. Lorenzini, P. Balatti, E. De Momi, and A. Ajoudani, “Pick the right co-worker: Online assessment of cognitive ergonomics in human-robot collaborative assembly,” *IEEE Transactions on Cognitive and Developmental Systems*, 2022.

[79] T. Segaran, *Programming collective intelligence: building smart web 2.0 applications*. ” O’Reilly Media, Inc.”, 2007.

[80] H. Wang, Y. Yu, and Q. Yuan, “Application of dijkstra algorithm in robot path-planning,” in *2011 second international conference on mechanic automation and control engineering*, IEEE, 2011, pp. 1067–1069.

[81] S. H. Semnani, A. H. de Ruiter, and H. H. Liu, “Force-based algorithm for motion planning of large agent,” *IEEE Transactions on Cybernetics*, 2020.

[82] A. Stoytchev, *Robot tool behavior: A developmental approach to autonomous tool use*. Georgia Institute of Technology, 2007.

[83] A. Z. Ren, B. Govil, T.-Y. Yang, K. R. Narasimhan, and A. Majumdar, "Leveraging language for accelerated learning of tool manipulation," in *Conf. on Robot Learning*, 2023, pp. 1531–1541.

[84] E. S. Reed, "Applying the theory of action systems to the study of motor skills," in *Advances in psychology*, vol. 50, Elsevier, 1988, pp. 45–86.

[85] D. E. McCoy, M. Schiestl, P. Neilands, R. Hassall, R. D. Gray, and A. H. Taylor, "New caledonian crows behave optimistically after using tools," *Current Biology*, vol. 29, no. 16, pp. 2737–2742, 2019.

[86] L. Jamone, "Modelling human tool use in robots," *Nature Machine Intelligence*, vol. 4, no. 11, pp. 907–908, 2022.

[87] H. Huang, C. Zeng, L. Cheng, and C. Yang, "Toward generalizable robotic dual-arm flipping manipulation," *IEEE Trans. on Industrial Electronics*, 2023.

[88] S. Y. Shin and C. Kim, "Human-like motion generation and control for humanoid's dual arm object manipulation," *IEEE Trans. on Industrial Electronics*, vol. 62, no. 4, pp. 2265–2276, 2014.

[89] X. Wu and Z. Li, "Cooperative manipulation of wearable dual-arm exoskeletons using force communication between partners," *IEEE Trans. on Industrial Electronics*, vol. 67, no. 8, pp. 6629–6638, 2019.

[90] M. Qin, J. Brawer, and B. Scassellati, "Robot tool use: A survey," *Frontiers in Robotics and AI*, vol. 9, p. 1 009 488, 2023.

[91] O. Kroemer, S. Niekum, and G. Konidaris, “A review of robot learning for manipulation: Challenges, representations, and algorithms,” *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 1395–1476, 2021.

[92] M. T. Mason, “Toward robotic manipulation,” *Annual Review of Control, Robotics, and Autonomous Syst.*, vol. 1, pp. 1–28, 2018.

[93] H.-Y. Lee, P. Zhou, B. Zhang, *et al.*, “A distributed dynamic framework to allocate collaborative tasks based on capability matching in heterogeneous multi-robot systems,” *IEEE Trans. on Cognitive and Developmental Syst.*, 2023.

[94] N. Wake, A. Kanehira, K. Sasabuchi, J. Takamatsu, and K. Ikeuchi, “Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration,” *arXiv preprint arXiv:2311.12015*, 2023.

[95] M. Ahn, A. Brohan, N. Brown, *et al.*, “Do as i can, not as i say: Grounding language in robotic affordances,” *arXiv preprint arXiv:2204.01691*, 2022.

[96] W. Huang, F. Xia, T. Xiao, *et al.*, “Inner monologue: Embodied reasoning through planning with language models,” *arXiv preprint arXiv:2207.05608*, 2022.

[97] I. Singh, V. Blukis, A. Mousavian, *et al.*, “Progprompt: Generating situated robot task plans using large language models,” in *IEEE Int. Conf. on Robotics and Automation*, 2023, pp. 11 523–11 530.

[98] J. Wu, R. Antonova, A. Kan, *et al.*, “Tidybot: Personalized robot assistance with large language models,” *arXiv preprint arXiv:2305.05658*, 2023.

[99] S. Huang, Z. Jiang, H. Dong, Y. Qiao, P. Gao, and H. Li, "Instruct2act: Mapping multi-modality instructions to robotic actions with large language model," *arXiv preprint arXiv:2305.11176*, 2023.

[100] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," *arXiv preprint arXiv:2303.12153*, 2023.

[101] M. Xu, P. Huang, W. Yu, *et al.*, "Creative robot tool use with large language models," *arXiv preprint arXiv:2310.13065*, 2023.

[102] T. Tsuji, J. Ohkuma, and S. Sakaino, "Dynamic object manipulation considering contact condition of robot with tool," *IEEE Trans. on Industrial Electronics*, vol. 63, no. 3, pp. 1972–1980, 2015.

[103] H. Wicaksono and C. Sammut, "Relational tool use learning by a robot in a real and simulated world," in *Proceedings of ACRA*, 2016.

[104] J. Brawer, M. Qin, and B. Scassellati, "A causal approach to tool affordance learning," in *IEEE Int. Conf. on Intelligent Robots and Syst.*, 2020, pp. 8394–8399.

[105] N. Saito, K. Kim, S. Murata, T. Ogata, and S. Sugano, "Tool-use model considering tool selection by a robot using deep learning," in *IEEE Int. Conf. on Humanoid Robots (Humanoids)*, 2018, pp. 270–276.

[106] P. Zech, S. Haller, S. R. Lakani, B. Ridge, E. Ugur, and J. Piater, "Computational models of affordance in robotics: A taxonomy and systematic classification," *Adaptive Behavior*, vol. 25, no. 5, pp. 235–271, 2017.

[107] K. P. Tee, J. Li, L. T. P. Chen, K. W. Wan, and G. Ganesh, “Towards emergence of tool use in robots: Automatic tool recognition and use without prior tool learning,” in *IEEE Int. Conf. on Robotics and Automation*, 2018, pp. 6439–6446.

[108] J. Sinapov and A. Stoytchev, “Detecting the functional similarities between tools using a hierarchical representation of outcomes,” in *IEEE Int. Conf. on Development and Learning*, 2008, pp. 91–96.

[109] S. Forestier and P.-Y. Oudeyer, “Modular active curiosity-driven discovery of tool use,” in *IEEE Int. Conf. on Intelligent Robots and Syst.*, 2016, pp. 3965–3972.

[110] S. Ding, J. Peng, J. Xin, H. Zhang, and Y. Wang, “Task-oriented adaptive position/force control for robotic systems under hybrid constraints,” *IEEE Trans. on Industrial Electronics*, 2024.

[111] H. Ochoa and R. Cortesão, “Impedance control architecture for robotic-assisted mold polishing based on human demonstration,” *IEEE Trans. on Industrial Electronics*, vol. 69, no. 4, pp. 3822–3830, 2021.

[112] M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, “Non-prehensile object transportation via model predictive non-sliding manipulation control,” *IEEE Trans. on Control Systems Technology*, 2023.

[113] M. B. Imtiaz, Y. Qiao, and B. Lee, “Prehensile and non-prehensile robotic pick-and-place of objects in clutter using deep reinforcement learning,” *Sensors*, vol. 23, no. 3, p. 1513, 2023.

[114] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, “Voxposer: Composable 3d value maps for robotic manipulation with language models,” *arXiv preprint arXiv:2307.05973*, 2023.

[115] T. Mar, V. Tikhanoff, G. Metta, and L. Natale, “Multi-model approach based on 3d functional features for tool affordance learning in robotics,” in *Int. Conf. on Humanoid Robots (Humanoids)*, IEEE, 2015, pp. 482–489.

[116] D. H. Douglas and T. K. Peucker, “Algorithms for the reduction of the number of points required to represent a digitized line or its caricature,” *Cartographica: the int. journal for geographic information and geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[117] A. Fowler and C. Wilson, “Cubic spline: A curve fitting routine,” Union Carbide Corp., Oak Ridge, Tenn. Y-12 Plant, Tech. Rep., 1966.

[118] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, “Density-based clustering,” *Wiley interdisciplinary reviews: data mining and knowledge discovery*, vol. 1, no. 3, pp. 231–240, 2011.

[119] T. Brown, B. Mann, N. Ryder, *et al.*, “Language models are few-shot learners,” *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[120] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig, “Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing,” *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

[121] M. Pragliola, L. Calatroni, A. Lanza, and F. Sgallari, “On and beyond total variation regularization in imaging: The role of space variance,” *SIAM Review*, vol. 65, no. 3, pp. 601–685, 2023.

[122] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, “Kpam: Keypoint affordances for category-level robotic manipulation,” in *The Int. Symposium of Robotics Research*, Springer, 2019, pp. 132–157.

[123] N. A. of Sciences, D. of Behavioral, B. on Science Education, *et al.*, *How people learn II: Learners, contexts, and cultures*. National Academies Press, 2018.

[124] J. Biggs, C. Tang, and G. Kennedy, *Teaching for quality learning at university 5e*. McGraw-hill education (UK), 2022.

[125] A. Sfard, “On two metaphors for learning and the dangers of choosing just one,” *Educational researcher*, vol. 27, no. 2, pp. 4–13, 1998.

[126] L. Jeffus, *Welding: principles and applications*. Cengage Learning, 2020.

[127] .

[128] L. Yasenko, Y. Klyatchenko, and O. Tarasenko-Klyatchenko, “Image noise reduction by denoising autoencoder,” in *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, IEEE, 2020, pp. 351–355.

[129] H. Yu, Q. Fu, Z. Yang, L. Tan, W. Sun, and M. Sun, “Robust robot pose estimation for challenging scenes with an rgb-d camera,” *IEEE Sensors Journal*, vol. 19, no. 6, pp. 2217–2229, 2018.

[130] S. Huo, B. Zhang, M. Muddassir, D. T. Chik, and D. Navarro-Alarcon, “A sensor-based robotic line scan system with adaptive roi for inspection of defects over convex free-form specular surfaces,” *IEEE Sensors Journal*, 2021.

[131] K. Wu, J. Tan, H. L. Xia, and C. B. Liu, “An exposure fusion-based structured light approach for the 3d measurement of a specular surface,” *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6314–6324, 2020.

[132] M. Purohit, M. Singh, A. Kumar, and B. K. Kaushik, “Enhancing the surveillance detection range of image sensors using hdr techniques,” *IEEE Sensors Journal*, vol. 21, no. 17, pp. 19 516–19 528, 2021.

[133] L. Yang, E. Li, T. Long, J. Fan, and Z. Liang, “A high-speed seam extraction method based on the novel structured-light sensor for arc welding robot: A review,” *IEEE Sensors Journal*, vol. 18, no. 21, pp. 8631–8641, 2018.

[134] L. Yang, E. Li, T. Long, J. Fan, and Z. Liang, “A novel 3-d path extraction method for arc welding robot based on stereo structured light sensor,” *IEEE Sensors Journal*, vol. 19, no. 2, pp. 763–773, 2018.

[135] Y. Tian, H. Liu, L. Li, *et al.*, “Automatic identification of multi-type weld seam based on vision sensor with silhouette-mapping,” *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5402–5412, 2020.

[136] M. Lorenzini, W. Kim, and A. Ajoudani, “An online multi-index approach to human ergonomics assessment in the workplace,” *IEEE Transactions on Human-Machine Systems*, 2022.

[137] M. Maniadakis, E. Hourdakis, and P. Trahanias, “Time-informed task planning in multi-agent collaboration,” *Cognitive Systems Research*, vol. 43, pp. 291–300, 2017.

[138] M. Wooldridge, *An introduction to multiagent systems*. John wiley & sons, 2009.

[139] P. Ghassemi, D. DePauw, and S. Chowdhury, “Decentralized dynamic task allocation in swarm robotic systems for disaster response,” in *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, IEEE, 2019, pp. 83–85.

[140] S. Amador, S. Okamoto, and R. Zivan, “Dynamic multi-agent task allocation with spatial and temporal constraints,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, 2014.

[141] B. Xie, J. Chen, and L. Shen, “Cooperation algorithms in multi-agent systems for dynamic task allocation: A brief overview,” in *2018 37th Chinese Control Conference (CCC)*, IEEE, 2018, pp. 6776–6781.

[142] Y. Han, D. Li, J. Chen, X. Yang, Y. Hu, and G. Zhang, “A multi-robots task allocation algorithm based on relevance and ability with group collaboration,” *International Journal of Intelligent Engineering and Systems*, vol. 3, no. 2, pp. 33–41, 2010.

[143] A. Clark, “Natural-born cyborgs?” In *International Conference on Cognitive Technology*, Springer, 2001, pp. 17–24.

[144] M. Beetz, D. Jain, L. Mosenlechner, *et al.*, “Cognition-enabled autonomous robot control for the realization of home chore task intelligence,” *Proceedings of the IEEE*, vol. 100, no. 8, pp. 2454–2471, 2012.

[145] M. Beetz, L. Mösenlechner, and M. Tenorth, "Cram—a cognitive robot abstract machine for everyday manipulation in human environments," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, 2010, pp. 1012–1017.

[146] D. Beßler, M. Pomarlan, and M. Beetz, "Owl-enabled assembly planning for robotic agents," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 1684–1692.

[147] S. Borgo, A. Cesta, A. Orlandini, and A. Umbrico, "Knowledge-based adaptive agents for manufacturing domains," *Engineering with Computers*, vol. 35, no. 3, pp. 755–779, 2019.

[148] R. Ros, S. Lemaignan, E. A. Sisbot, *et al.*, "Which one? grounding the referent based on efficient human-robot interaction," in *19th International Symposium in Robot and Human Interactive Communication*, IEEE, 2010, pp. 570–575.

[149] A. Sgorbissa, I. Papadopoulos, B. Bruno, C. Koulouglioti, and C. Recchiuto, "Encoding guidelines for a culturally competent robot for elderly care," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 1988–1995.

[150] M. Warnier, J. Guitton, S. Lemaignan, and R. Alami, "When the robot puts itself in your shoes. managing and exploiting human and robot beliefs," in *2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2012, pp. 948–954.

[151] Y. Chen, X. C. Ding, A. Stefanescu, and C. Belta, "Formal approach to the deployment of distributed robotic teams," *IEEE Transactions on Robotics*, vol. 28, no. 1, pp. 158–171, 2011.

[152] M. Fazel-Zarandi and M. S. Fox, "Inferring and validating skills and competencies over time," *Applied Ontology*, vol. 8, no. 3, pp. 131–177, 2013.

[153] I. Tiddi, E. Bastianelli, G. Bardaro, M. d'Aquin, and E. Motta, "An ontology-based approach to improve the accessibility of ros-based robotic systems," in *Proceedings of the knowledge capture conference*, 2017, pp. 1–8.

[154] L. Kunze, T. Roehm, and M. Beetz, "Towards semantic robot description languages," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 5589–5595.

[155] H. Zheng and Y. Wang, "A distributed framework for dynamic task allocation of multi-robot symbolic motion planning," in *2019 American Control Conference (ACC)*, IEEE, 2019, pp. 3291–3296.

[156] R. Zlot and A. Stentz, "Complex task allocation for multiple robots," in *Proceedings of the 2005 IEEE international conference on robotics and automation*, IEEE, 2005, pp. 1515–1522.

[157] S. C. Botelho and R. Alami, "M+: A scheme for multi-robot cooperation through negotiated task allocation and achievement," in *Proceedings 1999 IEEE international conference on robotics and automation (Cat. No. 99CH36288C)*, IEEE, vol. 2, 1999, pp. 1234–1239.

[158] A. M. Elmogy, A. M. Khamis, and F. O. Karray, "Market-based approach to mobile surveillance systems," *Journal of Robotics*, vol. 2012, 2012.

[159] J. Jimenez, A. Gomez, M. D. Buhrmester, A. Vázquez, H. Whitehouse, and W. B. Swann, "The dynamic identity fusion index: A new continuous measure of identity fusion for web-based questionnaires," *Social Science Computer Review*, vol. 34, no. 2, pp. 215–228, 2016.

[160] Z. Mandi, S. Jain, and S. Song, "Roco: Dialectic multi-robot collaboration with large language models," *arXiv preprint arXiv:2307.04738*, 2023.

[161] D. Seita, Y. Wang, S. J. Shetty, E. Y. Li, Z. Erickson, and D. Held, "Toolflownet: Robotic manipulation with tools via predicting tool flow from point clouds," in *Conf. on Robot Learning*, PMLR, 2023, pp. 1038–1049.

[162] X. Li, R. Cao, Y. Feng, *et al.*, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3961–3968, 2022.

[163] L. Jamone, E. Ugur, A. Cangelosi, *et al.*, "Affordances in psychology, neuroscience, and robotics: A survey," *IEEE Trans. on Cognitive and Developmental Syst.*, vol. 10, no. 1, pp. 4–25, 2016.

[164] M. Levihn and M. Stilman, "Using environment objects as tools: Unconventional door opening," in *IEEE Int. Conf. on Intelligent Robots and Syst.*, 2014, pp. 2502–2508.

[165] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning object affordances: From sensory–motor coordination to imitation," *IEEE Trans. on Robotics*, vol. 24, no. 1, pp. 15–26, 2008.

[166] J. Liang, W. Huang, F. Xia, *et al.*, "Code as policies: Language model programs for embodied control," in *IEEE Int. Conf. on Robotics and Automation*, 2023, pp. 9493–9500.

[167] H. Wang, G. Gonzalez-Pumariega, Y. Sharma, and S. Choudhury, "Demo2code: From summarizing demonstrations to synthesizing code via extended chain-of-thought," *arXiv preprint arXiv:2305.16744*, 2023.

[168] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Autotamp: Autoregressive task and motion planning with llms as translators and checkers," *arXiv preprint arXiv:2306.06531*, 2023.

[169] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," *arXiv preprint arXiv:2303.06247*, 2023.

[170] X. Zhang, Y. Zhu, Y. Ding, Y. Zhu, P. Stone, and S. Zhang, "Visually grounded task and motion planning for mobile manipulation," in *IEEE Int. Conf. on Robotics and Automation*, 2022, pp. 1925–1931.

[171] V. Sukhoy, V. Georgiev, T. Wegter, R. Sweidan, and A. Stoytchev, "Learning to slide a magnetic card through a card reader," in *Int. Conf. on Robotics and Automation*, 2012, pp. 2398–2404.

[172] M. Qin, J. Brawer, and B. Scassellati, "Rapidly learning generalizable and robot-agnostic tool-use skills for a wide range of tasks," *Frontiers in Robotics and AI*, vol. 8, p. 726 463, 2021.

[173] A. Stoytchev, "Computational model for an extendable robot body schema," *Georgia Institute of Technology, College of Computing, Tech. Rep. GIT-CC-03-44*, 2003.

[174] M. Shridhar, L. Manuelli, and D. Fox, "Perceiver-actor: A multi-task transformer for robotic manipulation," in *Conf. on Robot Learning*, PMLR, 2023, pp. 785–799.

[175] D. Driess, F. Xia, M. S. Sajjadi, *et al.*, “Palm-e: An embodied multimodal language model,” *arXiv preprint arXiv:2303.03378*, 2023.

[176] G. Courtadon, “A more accurate finite difference approximation for the valuation of options,” *Journal of Financial and Quantitative Analysis*, vol. 17, no. 5, pp. 697–703, 1982.

[177] C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, “Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping,” *IEEE Robotics and Automation Letters*, 2023.

[178] J. Park, S. Lim, J. Lee, *et al.*, “Clara: Classifying and disambiguating user commands for reliable interactive robotic agents,” *IEEE Robotics and Automation Letters*, 2023.

[179] M. Fuchs, “Homotopy equivalences in equivariant topology,” *Proceedings of the American Mathematical Society*, vol. 58, no. 1, pp. 347–352, 1976.

[180] M. Costanzo, M. De Simone, S. Federico, and C. Natale, “Non-prehensile manipulation actions and visual 6d pose estimation for fruit grasping based on tactile sensing,” *Robotics*, vol. 12, no. 4, p. 92, 2023.

[181] R. Gondokaryono, M. Haiderbhai, S. A. Suryadevara, and L. A. Kahrs, “Learning nonprehensile dynamic manipulation: Sim2real vision-based policy with a surgical robot,” *IEEE Robotics and Automation Letters*, 2023.