D 2026

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

COORDINATING HUMAN-ROBOT TEAMS IN MATRIX MANUFACTURING WITH MILP SCHEDULING AND AUGMENTED REALITY INTERFACES

GABRIEL DE MOURA COSTA

TESE DE DOUTORAMENTO APRESENTADA
À FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO EM
ENGENHARIA ELETROTÉCNICA E DE COMPUTADORES

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Coordinating Human-Robot Teams in Matrix Manufacturing with MILP Scheduling and Augmented Reality Interfaces

Gabriel de Moura Costa



Doctoral Program in Electrical and Computer Engineering

Advisor: Prof. Dr. Marcelo Roberto Petry

Second Advisor: Prof. Dr. António Paulo Gomes Mendes Moreira

October 16, 2026

© Gabriel de Moura Costa, 2026

Coordinating Human-Robot Teams in Matrix Manufacturing with MILP Scheduling and Augmented Reality Interfaces

Gabriel de Moura Costa

Doctoral Program in Electrical and Computer Engineering

Approved in oral examination by the committee:

President: Prof. Dr. Ana Cristina Costa Aguiar
Referee: Prof. Dr. Accacio Ferreira dos Santos Neto
Referee: Prof. Dr. Pedro Mariano Neto
Referee: Prof. Dr. Paulo Jorge Pinto Leitão
Referee: Prof. Dr. António Fernando Vasconcelos Cunha Castro Coelho
Referee: Prof. Dr. Paulo José Cerqueira Gomes da Costa
Advisor: Prof. Dr. Marcelo Roberto Petry

October 16, 2026

Resumo

A crescente procura por produtos altamente personalizados, aliada ao rápido avanço tecnológico, está a impulsionar a transição da produção em massa tradicional para sistemas de manufatura flexíveis e reconfiguráveis. Neste contexto, manter a eficiência enquanto se lida com elevada variabilidade de produtos, ciclos de produção mais curtos e eventos imprevistos tornou-se um desafio crítico. Para responder a estas exigências, é necessária uma coordenação eficaz entre recursos humanos e robóticos, tornando o escalonamento de equipas homem–robô em sistemas de manufatura matricial essencial para alcançar os níveis de flexibilidade, adaptabilidade e capacidade de resposta requeridos nos ambientes de produção modernos.

Em paralelo, as tecnologias de realidade aumentada têm vindo a afirmar-se como importantes facilitadores da Indústria 5.0, promovendo uma manufatura mais centrada no ser humano através de orientação contextual, informação em tempo real e interação intuitiva. Embora as técnicas de otimização para o escalonamento da produção e as interfaces baseadas em realidade aumentada tenham sido amplamente estudadas de forma independente, a sua aplicação conjunta em sistemas de manufatura colaborativos permanece relativamente pouco explorada.

Com o intuito de preencher essa lacuna, esta tese investiga a integração de um modelo de escalonamento baseado em otimização com uma interface de interação homem–máquina suportada por realidade aumentada num cenário colaborativo de desmontagem. Do ponto de vista da otimização, é proposto um conjunto de modelos de programação linear inteira mista para coordenar equipas homem–robô num sistema de manufatura matricial. O problema é decomposto em três etapas interligadas: gestão de baterias, alocação e escalonamento de tarefas, e relocalização de robôs móveis. Cada modelo incorpora restrições operacionais relevantes, incluindo capacidades dos agentes, tempos de transporte, disponibilidade de recursos e dinâmica dos níveis de bateria. Os modelos são resolvidos sequencialmente, sendo que as saídas de cada etapa alimentam a seguinte, permitindo uma tomada de decisão coordenada ao nível do sistema. A abordagem é validada através de uma implementação de prova de conceito num sistema de manufatura matricial de caráter industrial já implementado, demonstrando a sua capacidade de gerar soluções viáveis e otimizadas sob condições operacionais realistas.

Do ponto de vista da interação homem–robô, foram desenvolvidas e avaliadas experimentalmente duas interfaces homem–máquina distintas: uma interface industrial convencional baseada em ecrã tátil e uma aplicação de realidade aumentada implementada no HoloLens 2. Ambas seguem a mesma lógica instrucional para garantir comparabilidade, diferindo principalmente no modo de apresentação da informação: orientação baseada em vídeo e texto versus instruções holográficas espacialmente alinhadas com o ambiente físico. Foi conduzido um estudo experimental num cenário de desmontagem colaborativa, adotando uma abordagem de avaliação mista. Métricas objetivas de desempenho, como o tempo de execução e o número de erros, foram combinadas com avaliações subjetivas de usabilidade e carga cognitiva através de questionários validados (Escala de Usabilidade do Sistema e Índice de Carga de Trabalho da NASA), complementadas por questionários personalizados e entrevistas semiestruturadas. Esta metodologia permitiu uma avali-

ação abrangente do desempenho e da experiência do utilizador nas diferentes modalidades de interação.

Os resultados demonstram a viabilidade do quadro de escalonamento proposto na coordenação de atividades homem–robô num sistema de manufatura matricial, gerindo com sucesso a sequência de tarefas, a relocalização de robôs móveis e a gestão de baterias sob restrições operacionais realistas. O modelo conseguiu alocar tarefas de forma eficiente entre agentes humanos e robóticos, respeitando precedências e a disponibilidade de recursos. O domínio de aplicação validado do método corresponde a instâncias de pequena e média dimensão: por se tratar de uma abordagem exata, resolvida com *Branch-and-Cut*, o tempo de cálculo do módulo de escalonamento de tarefas cresce com o número de operações, agentes e estações, e a capacidade de reação do quadro está limitada aos pontos de indexação dos robôs móveis da plataforma utilizada. Estes limites definem a escala e o grau de dinamismo em que a abordagem pode ser aplicada diretamente, e orientam a sua extensão a sistemas de maior dimensão.

A avaliação das interfaces indica que a realidade aumentada melhora a clareza das instruções e aumenta a perceção das ações do robô por parte dos operadores, particularmente em cenários colaborativos. No entanto, a interface tátil obteve uma usabilidade significativamente superior e provocou menos fadiga ocular, enquanto a carga de trabalho global se revelou comparável entre as duas interfaces, sendo a eficiência da interação influenciada por fatores como os métodos de interação gestual, a ausência de feedback háptico e questões ergonómicas do dispositivo. Os resultados sugerem que a realidade aumentada apresenta particular potencial em contextos de formação e aprendizagem, enquanto interfaces táteis convencionais continuam a oferecer vantagens em operações rotineiras devido à sua eficiência e familiaridade para os utilizadores.

De forma geral, este trabalho contribui para o desenvolvimento de sistemas de manufatura mais flexíveis e centrados no ser humano através de duas contribuições principais. Em primeiro lugar, propõe uma estrutura de programação linear inteira mista que integra o escalonamento de tarefas, a relocalização de robôs móveis e a gestão de baterias, oferecendo uma abordagem sistemática e coordenada para otimizar o desempenho de equipes homem–robô em sistemas de manufatura matriciais. Em segundo lugar, investiga o design e a avaliação de uma interface homem–máquina baseada em realidade aumentada, comparando-a com uma interface convencional de ecrã tátil para identificar como a modalidade da interface influencia o desempenho do operador, a carga cognitiva e a compreensão das tarefas. Juntas, estas contribuições fornecem diretrizes práticas para o design de algoritmos de escalonamento e interfaces de operador que melhoram a eficiência operacional, suportam uma colaboração homem–robô intuitiva e orientam boas práticas para a implementação de sistemas de produção na Indústria 5.0.

Palavras-Chaves: Colaboração Homem–Robô, Sistemas de Manufatura Matricial, Escalonamento da Produção, Programação Linear Inteira Mista, Realidade Aumentada, Interfaces Homem – Máquina

Abstract

The increasing demand for highly customized products, together with rapid technological advancements, is driving the transition from traditional mass production to flexible and reconfigurable manufacturing systems. In this context, maintaining efficiency while coping with high product variability, shorter production cycles, and unforeseen disruptions has become a critical challenge. Addressing this challenge requires effective coordination of human and robotic resources, making the scheduling of human–robot teams in matrix manufacturing systems essential for achieving the flexibility, adaptability, and responsiveness expected in modern production environments.

At the same time, augmented reality technologies are increasingly recognized as key enablers of Industry 5.0, supporting human-centric manufacturing through contextual guidance, enhanced situational awareness, and intuitive interaction. Although optimization techniques for production scheduling and augmented reality-based human–machine interfaces have been widely studied, their combined application in collaborative manufacturing systems remains relatively unexplored.

To address this gap, this thesis investigates the integration of an optimization-based scheduling algorithm with an augmented reality-supported human–machine interaction interface in a collaborative disassembly scenario. From an optimization perspective, a set of mixed-integer linear programming models is proposed to coordinate human–robot teams within a matrix manufacturing system. The framework decomposes the problem into three interconnected stages: battery management, task allocation and scheduling, and mobile robot relocation. Each model captures key operational constraints, including agent capabilities, transportation times, resource availability, and battery-level dynamics. The models are solved sequentially, with outputs from one stage informing the next, enabling coordinated decision-making across the system. The approach is validated through a proof-of-concept implementation in an industry-deployed matrix manufacturing system, demonstrating its ability to generate feasible and optimized schedules under realistic operating conditions.

From a human–robot interaction perspective, two distinct human–machine interfaces were developed and experimentally evaluated: a conventional industrial touch screen interface and an augmented reality application implemented on the HoloLens 2. Both interfaces follow the same instructional logic to ensure comparability, differing primarily in the mode of information delivery: video and text-based guidance versus spatially registered holographic instructions. A user study was conducted in a collaborative disassembly task, adopting a mixed-method evaluation approach. Objective performance metrics, including task execution time and error rate, were combined with subjective assessments of usability and cognitive workload through standardized questionnaires (System Usability Scale and NASA-Task Load Index), complemented by custom questionnaires and semi-structured interviews. This methodology enabled a comprehensive evaluation of both performance and user experience across interaction modalities.

The results demonstrate the feasibility of the proposed scheduling algorithm in coordinating human–robot activities within a matrix manufacturing system, successfully handling task sequenc-

ing, mobile robot relocation, and battery management under realistic operational constraints. The framework efficiently allocated tasks to human and robotic agents while respecting precedence and resource availability. The validated scope of the method covers small- to medium-sized instances: as an exact approach solved with Branch-and-Cut, the computation time of the task scheduling module grows with the number of operations, agents, and workstations, and the responsiveness of the framework is bounded by the indexing points of the mobile robots of the platform used. These boundaries define the scale and degree of dynamism at which the approach can be applied directly, and guide its extension to larger systems.

The interface evaluation shows that augmented reality improves the clarity of instructions and enhances operators' awareness of robotic actions, particularly in collaborative scenarios. However, the touch screen interface achieved significantly higher usability and caused less eye fatigue, while overall workload was comparable between the two interfaces, and interaction efficiency was influenced by factors such as gesture-based input, lack of haptic feedback, and device ergonomics. The findings suggest that augmented reality is particularly beneficial for training and instructional contexts, whereas conventional touch interfaces remain advantageous for routine operations due to their efficiency and user familiarity.

Overall, this work advances the development of flexible and human-centric manufacturing systems through two main contributions. First, it proposes a mixed-integer linear programming framework that integrates task scheduling, mobile robot relocation, and battery management, providing a systematic and coordinated approach for optimizing human–robot team performance in matrix manufacturing systems. Second, it investigates the design and evaluation of an augmented reality-based human–machine interface, comparing it with a conventional touch screen interface to identify how interface modality affects operator performance, cognitive workload, and task understanding. Together, these contributions offer practical guidelines for designing scheduling algorithms and operator interfaces that improve operational efficiency, support intuitive human–robot collaboration, and inform best practices for the implementation of Industry 5.0 production systems.

Key-Words: Human–Robot Collaboration, Matrix Manufacturing Systems, Scheduling Optimization, Mixed-Integer Linear Programming, Augmented Reality, Human–Machine Interfaces

Acknowledgements

First and foremost, I thank God for granting me the opportunity to take part in this doctoral program, for strengthening me, and for guiding me with wisdom throughout this entire journey.

To the love of my life, Kaory, who endured with courage and patience the years of distance so that I could pursue the dream of this degree. Thank you for being by my side, even from afar, sharing endless nights of movies, series, and popcorn over video calls. Thank you for never giving up on us, for believing in our marriage — which, fortunately, came true even before the conclusion of this stage — and for dreaming with me every single day. You are my inspiration and the strongest, most resilient woman I know. I am grateful that you shared my dreams, no matter how challenging they were, and for allowing me to share in yours. Thank you for making me laugh when all I wanted was to cry and for always believing in me. I love you, today and always!

To my parents, Marco Antonio and Gislaine, for their unconditional encouragement, even from a distance. You are my role models in life and character, and an essential part of this achievement.

To the PACU (godparents and in-laws), my chosen families — Micaella and Davi; Rafael, Sabrie, and little Davi. To my friends from Brazil who remained present throughout these years: Antônio, Hygor, and Rafael Andrian.

To my colleagues at INESCTEC, who stood by me throughout this journey, enduring my changing moods between outbursts and jokes, and celebrating every milestone along the way. I am grateful for the long conversations, the shared coffees, the study nights, and above all, for being my support when I needed it most. A special thanks to João Costa, Catarina Rema, Marina Brilhante, Manuel Couto, and Ricardo Sousa, who were present not only until the end of my PhD but also at the beginning of my marriage. To João and Manuel, with humor and affection: we are better gymnasts than engineers!

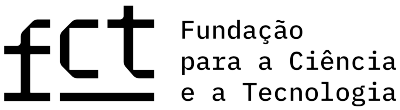
To my gymnastics friends and coaches, Simão Secca, Manuel Campos, among others, whose support helped me reach my best both physically and mentally.

To my family in Portugal, the Mello family: Aureliano and Josefina, Cristina, Luís, Rita and little Emma, as well as Fernanda and Richard, who welcomed me with open arms and supported me in every step of this journey. You adopted me, cared for me, and gave me strength as family would, and no words of gratitude will ever be enough to express how deeply thankful I am for all you have done.

Finally, I would like to thank Professors Marcelo Petry and António Paulo Moreira, who welcomed me with generosity and guided me with dedication throughout my academic journey. The lessons I learned from you during these more than five years go far beyond the academic sphere, shaping my professional and personal growth as well. It was an honor and a privilege to learn from you.

Gabriel de Moura Costa

Official Acknowledgements



This work was supported by Fundação para a Ciência e Tecnologia (FCT) within the scope of the PhD Grant 2022.10849.BD (the contract period ran from September 1, 2022, to August 31, 2026). DOI: https://doi.org/10.54499/2022.10849.BD.



Special thanks are extended to the Industry and Innovation Lab (iiLab) and the Center for Robotics in Industry and Intelligent Systems (CRIIS) at INESC TEC for hosting and supporting this research.

Publications Relevant to the Thesis

- **Costa, Gabriel M.**, Marcelo R. Petry, and António P. Moreira. "Augmented Reality for Human–Robot Collaboration and Cooperation in Industrial Applications: A Systematic Literature Review." *Sensors* 22, no. 7 (2022): 2725. DOI: https://doi.org/10.3390/s22072725
- **Costa, Gabriel M.**, Marcelo R. Petry, João G. Martins, and António P. Moreira. "Assessment of Multiple Fiducial Marker Trackers on HoloLens 2." *IEEE Access* 12 (2024): 14211–14226. DOI: https://doi.org/10.1109/ACCESS.2024.3356722
- Martins, João G., **Gabriel M. Costa**, Marcelo R. Petry, Paulo Costa, and António P. Moreira. "A Multi-User Multi-Robot Collaboration through Augmented Reality." In *2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 144–150. IEEE, 2024. DOI: https://doi.org/10.1109/ICARSC61747.2024.10535948
- **Costa, Gabriel M.**, Gonçalo Figueira, António P. Moreira, and Marcelo R. Petry. "A Robotic Coordination Framework for Human–Robot Teams in Matrix Manufacturing." *Applied Sciences* 16, no. 14 (2026): 7174. DOI: https://doi.org/10.3390/app16147174
- **Costa, Gabriel M.**, V. T. Santos, P. Alves, António P. Moreira, and Marcelo R. Petry. "Effects of Interface Modality on Human–Robot Collaboration: Comparing Immersive and Non-Immersive Solutions in Disassembly Tasks." *IEEE Access* (2026). DOI: https://doi.org/10.1109/ACCESS.2026.3744466

Contributions to Research Projects

The following R&D projects deployed this Ph.D. thesis proposal:

- **PRODUTECH 4S&C.** – "PRODUTECH SUSTENTÁVEL & CIRCULAR – Soluções inovadoras, sustentáveis e circulares com impacto na fileira das tecnologias de produção" (reference POCI-01-0247-FEDER-046102), co-financed by the European Regional Development Fund (ERDF) through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under the Portugal 2020 Partnership Agreement `http://mobilizadores.produtech.org/en/produtech-4-s-c`. The author was awarded a research grant under this project to conduct research in the areas of augmented reality and human–robot collaboration, contributing to the development of monitoring and interaction solutions, and participating in the design, implementation, and demonstration of collaborative robotics solutions and Industry 4.0 workstations in laboratory and industrial environments. The research conducted within this project resulted in the publication of scientific papers in peer-reviewed journals and international conferences.

- **iBot4CRMs.** European Union's Horizon Europe research and innovation programme under Grant Agreement No. 101189783. The author contributed by developing the Prolynk MMS system controller, which interfaced the Prolynk PLC with ROS and implemented an optimization algorithm for coordinating task distribution across the MMS, supporting assignment to individual agents or human–robot teams depending on station configuration. The system controller was integrated with INESC TEC's task manager to govern robotic manipulator actions. The author further developed both AR and conventional HMIs for the MMS, contributed to system documentation, troubleshooting, and participated in the preparation and execution of industrial demonstration activities.

- **NGS.** Component 5 – Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026, within project NGS, with reference 58. The author contributed by developing the interface between INESC TEC's Advanced Plant Model and the Prolynk MMS, through ROS-OPCUA communication between the MMS PLC and ROS. He further assisted in coordinating MMS operations with robotic manipulator actions via INESC TEC's task manager, developed gripper skills for Robotiq two-finger grippers on a UR5 using URCaps and TCP communication, and contributed to the implementation of a suction gripper skill.

- **MirrorLabs.** European Institute of Innovation and Technology (EIT) under grant agreement No. 20098, within the Horizon 2020 research and innovation programme. The author developed an AR interface for programming a UR5 robotic manipulator, incorporating a safety fence feature, calibration tools, and real-time communication between the AR glasses

and the ROS framework controlling the manipulator. The interface was validated in both simulated and real environments, integrated with INESC TEC's task manager, and accompanied by demonstration activities and instruction manuals.

Both religion and science require a belief in God. For believers, God is in the beginning, and for scientists He is at the end of all considerations... To the former He is the foundation, to the latter, the crown of the edifice of every generalized world view.

Max Planck (1858 – 1947)

For the Lord gives wisdom; From His mouth come knowledge and understanding.

Proverbs 2:6 (NKJV)

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 3
- 1.3 Research Questions . . . 4
- 1.4 Goals . . . 4
- 1.5 Thesis Contributions . . . 5
- 1.6 Thesis Outline . . . 6

2 Background **7**
- 2.1 Matrix Manufacturing Systems . . . 7
- 2.2 Production Scheduling and Mixed Integer Linear Programming . . . 9
- 2.3 Human-Robot Collaboration . . . 10
- 2.4 Augmented Reality . . . 13

3 Related Works **21**
- 3.1 Scheduling Human-Robot Teams in Matrix Manufacturing Systems . . . 21
  - 3.1.1 Flexible Job Shop Scheduling Problems . . . 22
  - 3.1.2 Human-Robot Teams Scheduling Problems . . . 23
- 3.2 Augmented Reality in Human-Robot Collaborative Assembly Scenarios . . . 26
  - 3.2.1 Interface Design and Operator Awareness in Human-Robot Collaboration 26
  - 3.2.2 Interface Modality in Scheduled and Dynamic Human-Robot Collaboration 30
- 3.3 Conclusion and Research Gaps . . . 32

4 System Architecture Description **35**
- 4.1 System Overview . . . 35
- 4.2 Matrix Manufacturing System . . . 36
- 4.3 Head-Mounted Display . . . 39
- 4.4 Tracking Algorithm . . . 40
  - 4.4.1 Methodology of the Comparative Study . . . 40
  - 4.4.2 Results and Discussion . . . 41
  - 4.4.3 Algorithm Selection and Implementation . . . 52
- 4.5 Communication Framework . . . 54

5 A Robotic Coordination Framework for Human-Robot Teams in Matrix Manufacturing Systems **57**
- 5.1 Problem Definition . . . 57
- 5.2 Proposed Coordination Pipeline . . . 58
- 5.3 Mathematical Formulations . . . 62

5.3.1 Battery Swap Formulation . . . 64
5.3.1.0.1 Variable domains and index convention. . . . 66
5.3.2 Task Scheduling Formulation . . . 70
5.3.2.0.1 Variable domains and index convention. . . . 71
5.3.3 Mobile Robot Relocation Formulation . . . 75
5.3.3.0.1 Variable domains and index convention. . . . 76
5.4 Coordination Framework Experimental Evaluation . . . 78
5.4.1 Use Case . . . 78
5.4.2 Coordination Framework Experimental Evaluation Results . . . 79
5.4.3 Discussion and Conclusions of the Coordination Framework . . . 83

6 Operator Interfaces for Human-Robot Collaboration in Matrix Manufacturing 87
6.1 Theoretical Framing: Situation Awareness, Robot Intent, and Attention . . . 89
6.2 Positioning with Respect to Prior Work . . . 91
6.2.1 Interface Configurations in Human-Robot Collaboration . . . 91
6.2.2 Collaborative Disassembly and Task Allocation . . . 92
6.3 Non-Immersive Interface . . . 92
6.4 Immersive Interface . . . 94
6.4.1 Design Rationale . . . 94
6.4.2 Technical Implementation . . . 95
6.4.3 Interface Structure and Human-Robot Interaction Relevant Features . . . 95
6.5 Differences Between the Interface Configurations . . . 97
6.6 Content Authoring and Extension to Other Products . . . 99
6.7 Experimental Validations and Results . . . 99
6.7.1 Study Design . . . 99
6.7.2 Study Protocol . . . 101
6.7.2.1 Participants . . . 101
6.7.2.2 Procedure . . . 103
6.7.3 Dependent Measures and Statistical Analysis . . . 109
6.7.3.1 Custom Questionnaire: Sources, Scales, and Reliability . . . 110
6.7.3.2 Statistical Analysis . . . 111
6.7.3.3 Interview Analysis . . . 112
6.7.4 Interfaces Results . . . 112
6.7.4.1 Overview of Primary Outcomes and Order Effects . . . 113
6.7.4.2 Human-Robot Interaction Outcomes . . . 113
6.7.4.2.1 Perceived Collaboration and Robot Awareness . . . 114
6.7.4.2.2 AR Visualization Quality . . . 116
6.7.4.3 Workload and Usability . . . 117
6.7.4.3.1 Eye-fatigue . . . 117
6.7.4.3.2 NASA-Task Load Index . . . 118
6.7.4.3.3 System Usability Scale . . . 119
6.7.4.3.4 General Task Experience . . . 122
6.7.4.4 Performance Indicators . . . 122
6.7.4.4.1 Error Analysis . . . 122
6.7.4.4.2 Execution Times . . . 126
6.7.4.4.3 Practice and Order Effects . . . 126
6.7.4.5 Exploratory Subgroup Analyses . . . 129
6.8 Discussion . . . 130
6.8.1 What the AR Configuration Improved, and Why . . . 130

6.8.2 Why the Same Configuration Increased Fatigue and Reduced Usability . 131
6.8.3 Comparison with Previous Studies 132
6.8.4 Practice Effects and Implications for Training 134
6.8.5 Implications for Interface Design in Industrial Human-Robot Collaboration 134
6.9 Limitations and Future Research 136
6.10 Operator Interfaces Conclusions 137

7 Conclusions and Future Work 139
7.1 Answers to the Research Questions 139
7.1.1 RQ1: Integrated MILP coordination in MMS 139
7.1.2 RQ2: Effect of the AR interface on user experience and performance . . 140
7.1.3 RQ3: Factors behind the differences in user experience and performance 140
7.2 Contributions 140
7.2.1 C1 – Knowledge synthesis on HRT scheduling and AR-supported HRC . 140
7.2.2 C2 – Benchmarking of fiducial marker tracking on the HL2 140
7.2.3 C3 – MILP-based robotic coordination framework for HRTs in MMSs . . 141
7.2.4 C4 – Comparative evaluation of immersive and non-immersive interfaces 141
7.3 Future Work 141
7.3.1 Direct Continuation of the Main Contributions 141
7.3.2 Methodological Improvements to the User Study 142
7.3.3 Extension of the System 143

References 145

A Questionnaires 163
A.1 Demographics Data Questionnaire 163
A.1.1 Demographics Questionnaire 163
A.1.2 Prior Experiences Questionnaire 163
A.2 System Usability Scale Questionnaire 164
A.3 NASA-Task Load Index Questionnaire 164
A.4 Custom Questionnaire 165
A.4.1 General Questions 165
A.4.2 Working with the Robot 165
A.4.3 Augmented Reality Visualization (AR Condition Only) 165
A.5 Interview Questions 166

B Detailed Statistical Results of the Operator Interface Study 167
B.1 Full Crossover Model Results and Internal Consistency 167
B.2 Exploratory Subgroup Screening 168
B.3 Exploratory Subgroup Follow-up 170
B.3.1 Prior Disassembly Experience and Workload 170
B.3.2 Prior Robot Experience and General Task Experience 170
B.4 Item-, Subscale-, and Operation-Level Analyses 175
B.5 Error Type and Location 177
B.6 Sequence-Specific Analyses 178

List of Figures

2.1 Industrial MMS implementations: (a) KUKA MMS concept; (b) Prolynk MMS. . 8
2.2 Overlapping workspaces of workers and robots. 11
2.3 Levels of HRI. 12
2.4 The XR continuum. 14

4.1 Overview of the Prolynk MMS. 37
4.2 Discharge profile of the MPC batteries. 38
4.3 Charge profile of the MPC batteries. 38
4.4 Microsoft HoloLens 2 HMD. 40
4.5 Architecture and setup of the fiducial marker experiment. 41
4.6 Spatial transformations between the HL2 and the fiducial marker. 42
4.7 Workspace of the fiducial marker experiment. 42
4.8 Vuforia translation and orientation errors. 44
4.9 ARToolKit translation and orientation errors. 46
4.10 ArUco translation and orientation errors. 47
4.11 QR Code translation and orientation errors. 49
4.12 Best-performing algorithms for translational estimation. 50
4.13 Best-performing algorithms for rotational estimation. 51
4.14 QR code used for pose estimation at the workstation. 53
4.15 Logical architecture of the system. 55

5.1 Flowchart of the coordination pipeline. 60
5.2 3D-printed car alternator used in the case study. 78
5.3 Experimental setup of the Prolynk MMS. 80
5.4 Gantt chart of the first scheduling cycle. 81
5.5 MPC station assignments across the ten scheduling cycles. 84

6.1 Pages of the touch screen HMI. 94
6.2 Components of the AR HMI. 97
6.3 Age distribution of participants. 102
6.4 Gender distribution of participants. 103
6.5 Educational background of participants. 104
6.6 Fields of study of participants. 104
6.7 Occupational status of participants. 105
6.8 Professional roles of participants. 105
6.9 General use of eyesight correction. 106
6.10 Use of eyesight correction during the experiment. 107
6.11 Prior experience with AR/VR. 108
6.12 Prior experience with disassembly tasks. 108

6.13 Prior experience working with robots. 109
6.14 Mean Working with the Robot scores by interface. 115
6.15 Mean Working with the Robot item scores by interface. 115
6.16 Mean AR Visualization item scores. 117
6.17 Mean eye fatigue by interface. 118
6.18 Mean NASA-TLX scores by interface. 119
6.19 Mean NASA-TLX component scores by interface. 120
6.20 Mean SUS scores by interface. 121
6.21 Mean SUS item scores by interface. 121
6.22 Mean General Questions scores by interface. 122
6.23 Mean General Questions item scores by interface. 123
6.24 Participants with and without errors by interface. 123
6.25 Total error occurrences. 124
6.26 Mean number of errors per participant by interface. 125
6.27 Errors by type. 125
6.28 Errors by operation. 126
6.29 Errors by operation and type. 127
6.30 Mean execution and procedure times by interface. 128
6.31 Mean execution time per operation by interface. 129
6.32 Mean execution and procedure times by first interface. 130

B.1 Mean NASA-TLX scores by disassembly experience. 172
B.2 Mean NASA-TLX component scores by disassembly experience. 173
B.3 Mean General Questions item scores by robot experience. 174
B.4 Mean General Questions scores by robot experience. 175
B.5 Mean execution time per operation by first interface. 179
B.6 Mean number of errors by first interface. 180
B.7 Error types by first interface: frequencies. 181
B.8 Errors per operation by first interface: frequencies. 182

List of Tables

1.1 Research gaps, questions, contributions, chapters, and publications. 5

3.1 Overview of key studies on the FJSSP. 24
3.2 Overview of key studies on HRT scheduling problems. 25
3.3 Overview of AR-based HRC studies without scheduling algorithms. 28
3.4 Overview of AR-based HRC studies in scheduled collaboration contexts. 31

4.1 Mean translation and rotation errors of Vuforia. 43
4.2 Mean translation and rotation errors of ARToolKit. 45
4.3 Mean translation and rotation errors of ArUco. 48
4.4 Mean translation and rotation errors of QR Code. 50
4.5 Qualitative comparison of the four marker tracking algorithms. 52
4.6 Implementation of the architecture components. 55

5.1 Objectives, inputs, constraints, and outputs of the three modules. 59
5.2 Subscripts used in the MILP formulations. 63
5.3 Sets used in the MILP formulations. 63
5.4 Decision variables of the Battery Swap module. 65
5.5 Parameters of the Battery Swap module. 65
5.6 Decision variables of the Task Scheduling module. 71
5.7 Parameters of the Task Scheduling module. 72
5.8 Decision variables of the Mobile Robot Relocation module. 76
5.9 Parameters of the Mobile Robot Relocation module. 76
5.10 Operations, components, and admissible agents of the case study. 79
5.11 Transportation times between workstations and the ABC. 79
5.12 Transportation times between slots of the ABC. 80
5.13 Task allocation and execution times: first scheduling cycle. 80
5.14 Computational performance of the first scheduling cycle. 81
5.15 Task allocation and execution times: second scheduling cycle. 82
5.16 Computational performance of the second scheduling cycle. 82
5.17 MPC station assignments across scheduling cycles. 83
5.18 Comparison of the coordination framework with related works. 85

6.1 Mapping between theoretical constructs, interface features, and measures. 90
6.2 Shared elements and differences between the two interface configurations. 98
6.3 Components, sequence, and agent types of the alternator disassembly. 100
6.4 Primary outcomes and crossover mixed-model estimates. 113
6.5 Working with the Robot items: paired comparisons. 114
6.6 Execution time by operation: interface and session effects. 128

6.7 Comparison of the present user study with the closest AR-based HRC studies. . . 133
6.8 Design guidelines for operator interfaces in industrial HRC. 135

B.1 Primary outcomes: full crossover model results. 167
B.2 Internal consistency of the questionnaire blocks (Cronbach's α). 168
B.3 Working with the Robot scores by prior experience group. 168
B.4 AR Visualization scores by prior experience group. 168
B.5 Eye fatigue by use of visual correction. 168
B.6 NASA-TLX scores by prior experience group. 169
B.7 SUS scores by prior experience group. 169
B.8 General Questions scores by prior experience group. 169
B.9 Errors by prior experience group. 169
B.10 Execution and procedure times by prior experience group. 170
B.11 NASA-TLX scores by disassembly experience. 170
B.12 NASA-TLX components by disassembly experience. 171
B.13 General Questions scores by robot experience. 171
B.14 General Questions items by robot experience. 172
B.15 NASA-TLX components: group analysis. 175
B.16 SUS items: group analysis. 176
B.17 General Questions items: group analysis. 176
B.18 Execution times per operation: group analysis. 176
B.19 Error types by interface (Fisher's exact test). 177
B.20 Errors per operation by interface (Fisher's exact test). 177
B.21 Global chi-square test of the error distribution. 177
B.22 Execution and procedure times by first interface. 178
B.23 Execution times per operation by first interface. 178
B.24 Errors by first interface. 179
B.25 Error types by first interface. 179
B.26 Errors per operation by first interface. 180

List of Acronyms

3D Three-Dimensional

C_{max} Makespan

ABC Automatic Battery Changer

ABCOA Artificial Bee Colony Optimization Algorithm

ACO Ant Colony Optimization

AGV automated guided vehicle

AOG AND/OR Graph

AR Augmented Reality

AV Augmented Virtuality

B&B Branch and Bound

BT Behavior Trees

CP Constraint Programming

CPPS Cyber-Physical Production System

DSP Disassembly Sequence Planning

DT Digital Twins

FIFO First-In First-Out

FJSSP Flexible Job Shop Scheduling Problem

GA Genetic Algorithm

GRASP Greedy Randomized Adaptive Search Procedure

HFOA Hybrid Fruit Fly Optimization Algorithm

HL HoloLens

HMD Head-Mounted Display

HMI Human-Machine Interface

HRC Human-Robot Collaboration

HRI Human-Robot Interaction

HRT Human-Robot Team

HTLBOA Hybrid Teaching and Learning-Based Optimization Algorithm

IAR Industrial Augmented Reality

IGA Iterated Greedy Algorithm

IMU Inertial Measurement Unit

ISO International Organization for Standardization

KUKA Keller und Knappich Augsburg

LDILS Learning-Driven Iterated Local Search Algorithm

LLM Large Language Model

M2SLHO Multi-Micro-Swarm Leadership Hierarchy-Based Optimization Algorithm

MA Memetic Algorithm

MA-A2C Multi-Agent Advantage Actor-Critic

MA-TRL Multi-Agent Transfer Reinforcement Learning

MES Manufacturing Execution System

MILP Mixed Integer Linear Programming

ML Machine Learning

MMS Matrix Manufacturing System

MMTSP Multi-Mode Task Scheduling Problem

MPC Programmable Mobile Cobot

MR Mixed Reality

NSGA II Non-Dominated Sorting Genetic Algorithm II

MRLLD Multi-Agent Reinforcement Learning-enhanced Lagrangian Decomposition

MRTK Mixed Reality Toolkit

MTCP-SPEA multi-threaded constraint programming assisted space perception evolutionary algorithm

NASA-TLX National Aeronautics and Space Administration - Task Load Index

OPC-UA Open Platform Communications – Unified Architecture

PLC Programmable Logic Controller

PEDA Pareto-based Estimation of Distribution Algorithm

pHRI Physical Human-Robot Interaction

PT net Petri Net

PSO Particle Swarm Optimization

QR Quick Response

RL Reinforcement Learning

ROS Robot Operating System

RULA Rapid Upperlimb Assessment

SA Situation Awareness

SAHA Self-Adaptive Hybrid Algorithm

SDK Software Development Kit

SLAM Simultaneous Localization and Mapping

SLMA Self-Learning Memetic Algorithm

SL-VNS Self-learning Variable Neighborhood Search

SSQ Simulation Sickness Questionnaire

SUS System Usability Scale

UR Universal Robots

UX User Experience

VLM Vision Language Model

VR Virtual Reality

XR Extended Reality

WS Workstation

Chapter 1

Introduction

1.1 Context

The emergence of the Industry 4.0 and Industry 5.0 paradigms reflects the ongoing transition from traditional mass production toward highly flexible, interconnected, and increasingly customized manufacturing systems [1]. Industry 4.0, introduced in 2011 as part of the German government's high-tech strategy, represents the Fourth Industrial Revolution and is centered on the development of Cyber-Physical Production Systems (CPPSs). These systems enable intelligent, communication and decentralized decision-making, supporting efficient and flexible production of high-quality, customized products. This paradigm is driven by key enabling technologies, including the industrial internet of things, cloud computing, big data analytics, simulation, Augmented Reality (AR), autonomous robots, and system integration [2], [3].

Its core design principles encompass interoperability, adaptability, data integration across the product lifecycle, service-oriented architectures, and secure communication. While Industry 4.0 is predominantly technology- and productivity-driven, aiming to enhance efficiency and global competitiveness, it also introduces considerations related to sustainability, resilience, and human involvement. These include resource-efficient production, adaptive supply chains, and an increasing emphasis on Human-Robot Collaboration (HRC) supported by assistive technologies such as AR [2], [3].

Building upon this foundation, Industry 5.0, formally introduced by the European Commission in 2021, represents a shift from a purely technology-driven paradigm to a value-oriented approach. It complements Industry 4.0 by aligning technological innovation with broader societal and environmental objectives, particularly in response to challenges such as sustainability, supply chain vulnerabilities, and evolving workforce dynamics [3], [4].

At its core, Industry 5.0 is structured around three key principles: human-centricity, sustainability, and resilience. It emphasizes placing worker well-being at the center of production systems, promoting circular and resource-efficient processes, and ensuring robustness against disruptions. In contrast to Industry 4.0, it prioritizes the integration of human capabilities with intelligent systems, fostering adaptive HRC and continuous upskilling [3], [4].

To support this vision, Industry 5.0 leverages enabling technologies such as artificial intelligence, digital twins, advanced data infrastructures, Human-Robot Interaction (HRI) technologies, and energy-efficient solutions. Ultimately, it envisions a collaborative, sustainable, and resilient industrial ecosystem in which technology serves human and societal needs rather than acting as the sole driver of transformation [3], [4].

In alignment with these paradigms, and in response to increasing market dynamism and unforeseen disruptions, the matrix manufacturing concept has emerged to address requirements for enhanced flexibility and adaptability [5], [6]. Matrix Manufacturing Systems (MMSs) are characterized by modular, function-oriented workstations that replace fixed, linear production lines, enabling dynamic product routing and rapid system reconfiguration. In this context, manufacturing routes are not predefined but are determined in real time based on resource availability, transport times, system disturbances, and operator presence. Within this context, the integration of Human-Robot Teams (HRTs) fosters a synergistic collaboration across multiple levels of interaction. Such collaboration supports flexible and efficient manufacturing processes by combining the strength, precision, repeatability, and endurance of collaborative robots with the dexterity, adaptability, and cognitive capabilities, such as creativity, decision-making, and problem-solving, of human operators [7]–[9].

The transition from purely manual or fully automated production systems to human–robot collaborative environments must create value not only for manufacturing operations, such as reduced processing times and improved product quality, but also for operators, by improving ergonomics and reducing physical and cognitive workload [9]. In this context, Romero, Bernus, Noran, *et al.* [10] and Romero, Stahre, Wuest, *et al.* [11] introduced the concept of Operator 4.0 from a human-centered perspective within Industry 4.0, defining it as an intelligent and skilled worker supported by machines when needed. Building on this, Zolotová, Papcun, Kajáti, *et al.* [12] further consolidated the notion through the concept of the "smart operator," described as a worker who leverages advanced technologies, such as augmented and virtual reality, wearable devices, and data-driven tools, to interact with production systems, interpret context-sensitive information, and collaborate with robots, thereby enhancing decision-making and overall operational effectiveness.

Extending this vision, the concept of the Resilient Operator 5.0 describes an adaptable and resourceful employee who applies creativity, expertise, and technological advancements to challenge conventional practices and drive economic innovations, ensuring the continuity and success of operations even under unforeseen or challenging circumstances. In increasingly dynamic and flexible manufacturing environments, this role goes beyond merely executing predefined tasks, positioning the operator as an active decision-making agent. Operators must interpret contextual information, coordinate with robotic systems, and adapt to continuously changing production conditions and system decisions. This evolution emphasizes the importance of effective human–robot teaming and underscores the need for intuitive, context-aware interfaces that support operators in complex, adaptive, and resilient manufacturing scenarios [13], [14].

Consequently, research efforts and commercial advancements in enabling technologies such as AR have increased substantially. Advanced AR devices, including the Microsoft HoloLens (HL)

2, are being investigated not only as innovative tools but also as practical solutions to address a concrete challenge in manufacturing: the visualization and execution of task instructions at the workstation [9], [15]–[17]. In environments characterized by higher variability and increased HRC, AR interfaces move beyond a peripheral role and become an integral part of operational performance, directly supporting operators in interpreting contextual information, coordinating actions, and executing tasks efficiently. This highlights that the flexibility of MMS depends not only on the global coordination of resources but also on the availability of effective, context-aware interfaces that enable HRT to function seamlessly, motivating the present investigation into AR-supported operator interaction within adaptive production settings.

1.2 Motivation

For industries to effectively adopt AR applications in cooperative and collaborative tasks, future developments must increasingly rely on end-user experience and feedback. Such user-centered approaches are considered essential for further enhancing HRC features, particularly with respect to multi-agent communication and interaction among operators, devices, and services. Strengthening these aspects is expected to promote more dynamic and seamless information exchange across manufacturing systems [9], [18], [19].

To fully exploit the potential of MMS and HRTs, advanced scheduling algorithms play a critical role as decision-support tools. These algorithms optimize task allocation and sequencing while ensuring efficient resource utilization under multiple production constraints [6], [7], [20], [21], In MMS, where product routes are not predefined and decisions must be made in real time based on the availability of stations and operators, transport times, and unexpected disruptions, the integration of scheduling algorithms with AR-based interfaces becomes essential. AR interfaces provide operators with context-sensitive information about task assignments, system status, and collaborative actions with robots. This enables operators to interpret dynamic changes and act accordingly [9], [19], [22].

Despite significant advances in optimization research addressing task allocation within HRTs [7], [20], [23], most existing approaches focus on isolated collaborative workstations and do not integrate HRT allocation within the broader scheduling framework of MMS [21], [24]. Conversely, in the AR research domain, only a limited number of studies have evaluated the combined impact of AR interfaces and HRT scheduling algorithms [9], [25], [26], and most contributions remain confined to controlled or academic settings, which limits their industrial applicability [18], [22], [27].

The approach proposed in this thesis integrates advanced scheduling algorithms with AR-supported HRI to improve both manufacturing performance and operator experience. By providing context-sensitive information, the system enables operators to make informed decisions, coordinate effectively with robots, and adapt to dynamic production conditions. This not only enhances flexibility and resilience in MMS but also reduces cognitive and physical workload, supporting

the operator as a “smart” and empowered participant, in line with the human-centric principles of Industry 5.0.

1.3 Research Questions

Building on this motivation, this thesis aims to extend current knowledge by integrating HRT considerations into MMS scheduling models and experimentally evaluating the impact of AR-based Human-Machine Interface (HMI) compared to conventional industrial interfaces within a realistic production setting. The study is guided by the following research questions:

- RQ1: How effectively do integrated Mixed Integer Linear Programming (MILP) formulations for task scheduling, robot relocation, and battery management support efficient and coordinated operation in MMSs?
- RQ2: To what extent does the AR interface improve user experience and performance compared to a conventional HMI?
- RQ3: Which factors contribute to the improvement or deterioration of user experience and performance?

1.4 Goals

The overarching scientific goal of this thesis is to advance the understanding of how HRC can be systematically coordinated in flexible MMS while simultaneously enhancing operator support through AR-based interfaces. The research focuses on the interplay between optimization-driven task allocation at the system level and context-aware, human-centric interfaces at the workstation level. It aims to provide both methodological insights and empirical evidence on effective human–robot teaming in dynamic manufacturing environments.

To achieve this overarching goal, the thesis pursues the following specific scientific objectives:

- Critically review and synthesize the state-of-the-art in HRT scheduling and AR-supported HRC, identifying research gaps that inform the formulation of integrated models and experimental investigations in this thesis.
- Formulate and implement MILP-based scheduling models for HRTs in MMSs, capturing dynamic interactions among tasks, resources, and industrial robots. The contribution lies in the formal problem definition, integration of HRT constraints, and demonstration of model feasibility in realistic scenarios.
- Experimentally evaluate AR-supported interfaces versus conventional industrial touch screen HMI in an industrial disassembly setting, providing quantitative and qualitative evidence on their impact on operator decision-making, usability, and operational performance.

1.5 Thesis Contributions

This thesis advances the state of the art in HRC within flexible MMS by integrating system-level scheduling with operator-level support through AR-based interfaces. Its two main scientific contributions are the coordination framework (C3) and the comparative interface evaluation (C4); the remaining two contributions support them:

- **C1 – Knowledge synthesis:** a structured review of HRT scheduling and AR-supported HRC, identifying the gaps addressed in this thesis (Chapter 3).
- **C2 – Benchmarking of fiducial marker tracking on the HL2:** a comparison of four tracking algorithms against OptiTrack ground truth, which supports the tracking choice of the AR interface (Section 4.4).
- **C3 – MILP-based robotic coordination framework:** three sequentially solved MILP modules for battery swapping, task scheduling, and mobile robot relocation of HRTs in an MMS, validated on a deployed industrial platform (Chapter 5).
- **C4 – Comparative evaluation of immersive and non-immersive interfaces:** a counterbalanced user study comparing an AR interface with a functionally matched industrial touch screen HMI in collaborative disassembly, from which design guidelines for interfaces in industrial HRC are derived (Chapter 6).

Table 1.1 relates the research gaps, research questions, contributions, chapters, and publications of this thesis.

Table 1.1: Relation between research gaps, research questions, contributions, chapters, and publications.

Research gap	**RQ**	**Contribution**	**Chapter**	**Publication**
Task allocation, mobile robot logistics, battery management, and collaborative execution modes are treated in isolation; no integrated framework for HRTs in an MMS	RQ1	C1, C3	3, 4, 5	[21]
AR rarely compared with an equivalent industrial HMI; HRI-specific measures (robot awareness, perceived collaboration) rarely used; AR in scheduling-driven production little explored	RQ2, RQ3	C1, C4	3, 6	[9], [22]
Choice of tracking algorithm for spatial registration on the HL2 lacks an empirical basis	–	C2	4	[17]

1.6 Thesis Outline

The remainder of this thesis is organized as follows. Chapter 2 provides the fundamental background and key concepts supporting this work. It introduces the principles of MMS, production scheduling and MILP, HRC, and AR technologies, laying the conceptual foundation for understanding both system-level coordination and operator-level interaction in flexible manufacturing environments.

Chapter 3 presents a critical review of the literature on HRT scheduling in MMS and the application of AR in HRC contexts. The chapter identifies research gaps, evaluates existing methodologies, and motivates the proposed integration of optimization algorithms with human-centered interfaces. This review establishes the scientific context for the contributions of this thesis.

Chapter 4 details the system and communication architecture implemented in this research. It describes the MMS hardware and software components, the AR interface, and the tracking and communication framework that enables coordination among operators, robots, and the scheduling system. This chapter highlights the technical foundation for integrating operational planning and human–machine interaction.

Chapter 5 presents the proposed optimization framework. It defines the scheduling problem for HRTs in MMS, describes the MILP formulations for task scheduling, robot relocation, and battery management, and demonstrates the feasibility of the proposed approach. This chapter addresses the system-level contribution of the thesis by formalizing how dynamic, flexible production can be coordinated under real constraints.

Chapter 6 focuses on the operator interfaces. It describes the conventional touch screen and the AR-based interface, their features, interaction paradigms, and differences, and presents the user study that compares them in a collaborative disassembly task, including its design, statistical analysis, results, and discussion. The chapter concludes with design guidelines for interfaces in industrial HRC.

Finally, Chapter 7 concludes the thesis. It summarizes the main findings, addresses the research questions, and outlines directions for future work, including opportunities to extend the integration of human-centered interfaces with adaptive scheduling in increasingly flexible and resilient manufacturing systems.

Chapter 2

Background

This chapter introduces the concepts and terminology on which the rest of the thesis builds; it defines what the main concepts are, whereas the analysis of specific studies and the identification of research gaps are left to Chapter 3. The research is structured around four complementary domains, each addressing a distinct level of the proposed approach. MMS define the production context and system-level perspective, enabling flexible and dynamic allocation of resources in response to changing conditions. Production scheduling and MILP provide the decision-making tools used to coordinate this environment. HRC establishes the operational model, describing how human operators and robotic systems interact and coordinate tasks within this environment. AR, in turn, represents the interface and operator-support layer, providing context-aware information that enables operators to effectively interpret system decisions and collaborate with robots.

Together, these domains form a coherent conceptual framework that connects system-level coordination with operator-level interaction, which is central to the contributions of this thesis. Section 2.1 introduces the concept of MMS, Section 2.2 presents the scheduling and optimization concepts used in this work, Section 2.3 discusses HRC, and Section 2.4 presents AR technologies, thereby preparing the ground for the methodologies and approaches developed in the subsequent chapters.

2.1 Matrix Manufacturing Systems

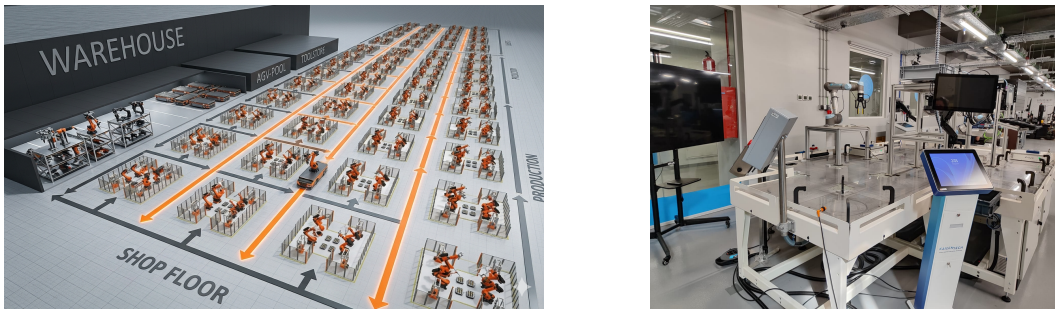
The linear, sequential, and standardized production lines that characterized the third industrial revolution, prioritizing high productivity and continuous product flow, no longer fully meet today's consumer expectations for customization and personalization [28]–[30]. To remain competitive while addressing this growing demand for innovative and tailored products, industries must sustain high production volumes without compromising quality. Achieving this balance requires enhanced production flexibility, decentralized decision-making, and intelligent utilization of available resources, thereby enabling mass customization without substantially increasing operational costs [18], [28], [29].

In response to these emerging requirements, new manufacturing paradigms have been proposed, among which the concept of MMS has gained particular relevance. An MMS consists of a network of autonomous workstations arranged in a grid-like topology, designed to ensure modularity, scalability, and mobility. These workstations are interconnected by a fleet of mobile platforms that enable flexible material flow throughout the system. Each workstation operates as a reconfigurable, product-neutral production unit capable of performing one or more standardized value-adding operations. The inclusion of partially redundant workstations further enhances system flexibility, allowing the production of highly individualized products while accommodating varying and independent cycle times within the same manufacturing environment [5], [6], [31], [32].

From an operational perspective, these characteristics fundamentally change how production is planned and executed. By decoupling logistics from production, MMS enables dynamic routing of jobs, where processing sequences are not fixed but determined in real time based on system conditions. As a result, production must handle increased variability in processing and transport times, while continuously allocating tasks to available resources, including machines, mobile platforms, and human operators. This leads to a highly dynamic decision-making environment, where scheduling becomes a continuous and adaptive process rather than a static, pre-defined plan [6].

While these features provide significant advantages in terms of flexibility and responsiveness, they also introduce important challenges. The absence of predefined routes and the presence of multiple alternative resources increase the complexity of planning and control. Effective coordination is required among workstations, mobile platforms, and operators, while system performance becomes sensitive to factors such as transport delays, machine failures, and resource availability. Consequently, ensuring efficient operation in MMS requires advanced decision-support mechanisms capable of handling information and multiple interdependent constraints [33].

Industrial implementations of MMS at different scales are presented by Keller und Knappich Augsburg (KUKA) [34] and Prolynk [35], illustrated in Figure 2.1.



(a) KUKA MMS concept. Adapted from [34].

(b) Prolynk MMS.

Figure 2.1: Industrial MMS implementations: (a) KUKA concept, with a modular grid of workstations connected through mobile platforms that enable non-sequential production flows; (b) Prolynk MMS, with a modular platform-based layout and mobile transport units that route products between heterogeneous workstations.

The Prolynk system is the experimental platform of this thesis. It is built on a grid of standardized platform units onto which manual assembly stations, robotic cells, and process-specific units (e.g., screwing, inspection, or testing) can be integrated. Products travel between workstations on a fleet of Programmable Mobile Cobots (MPCs), mobile platforms that follow the matrix layout and allow operations to be executed in any feasible order, while a fleet-management and control layer tracks the mobile units and communicates with industrial controllers, such as a Programmable Logic Controller (PLC), and with higher-level decision systems. Operators interact with manual and automated stations and share workspaces with robots, so that tasks can be assigned to operators or robots depending on system conditions. At the same time, operators must follow dynamically assigned tasks and coordinate with mobile platforms and robots, which highlights the need for effective decision-support interfaces. The specific configuration used in this work is described in Section 4.2.

2.2 Production Scheduling and Mixed Integer Linear Programming

Production scheduling is the allocation of limited resources to tasks over time so as to optimize one or more objectives [36]. In the standard terminology, a *job* is a product or order to be processed, an *operation* is one processing step of a job, and a *machine* (or, more generally, a resource) is the entity that performs it. Precedence constraints define the order in which the operations of a job must be executed, and the problem is *non-preemptive* when an operation, once started, cannot be interrupted. A *schedule* specifies which resource executes each operation and when.

Two families of problems are central to this thesis. In the classical job shop, each operation can be processed by a single, predefined machine; the Flexible Job Shop Scheduling Problem (FJSSP) relaxes this assumption by allowing each operation to be processed by any machine from a set of eligible machines, so that the problem combines a routing decision (which machine) with a sequencing decision (in which order) [37], [38]. In multi-mode scheduling problems, an operation may be executed in several alternative modes, each requiring a different combination of resources and taking a different processing time; in the multiprocessor variant, a mode may require several resources simultaneously [39], [40]. This is the structure of human-robot team scheduling, in which an operation may be executed by a human, by a robot, or jointly by both [7].

Scheduling objectives are usually time-based. The *makespan* (C_{max}) is the completion time of the last operation, i.e., the time needed to complete all the work, and is the most common objective in the FJSSP literature [37]. The *flowtime* (or total completion time) is the sum of the completion times of all operations or jobs; minimizing it favors finishing each operation as early as possible, and it is therefore often used as a secondary criterion. Other objectives include tardiness, which measures lateness with respect to due dates, and cost- or energy-based criteria [24]. When several objectives must be considered, they can be combined in a weighted sum or optimized *lexicographically*, i.e., in a predefined order of priority in which each objective is optimized without degrading the optimal values of the higher-priority ones [41].

A common way to formulate scheduling problems is MILP. A MILP model optimizes a linear objective function subject to linear constraints, with some decision variables restricted to integer values [42]. In scheduling, binary variables typically represent discrete decisions (e.g., whether an operation is assigned to a given resource, or whether one operation precedes another), and continuous variables represent times (e.g., start and end times). Logical conditions of the type *"if operation a precedes operation b on the same resource, then b starts after a ends"* are expressed as linear constraints with the *big-M* technique, in which a sufficiently large constant M multiplied by a binary variable deactivates the constraint when the condition does not hold [42]. Big-M formulations are flexible and easy to extend, but large values of M weaken the bounds that the solver can compute, which increases solution times.

MILP models are solved by exact methods that guarantee optimality. Branch and Bound (B&B) recursively partitions the problem into subproblems, solving the linear relaxation of each (i.e., the problem without integrality restrictions) to obtain bounds that allow subproblems that cannot contain a better solution to be discarded. *Branch-and-Cut* extends B&B by adding cutting planes, i.e., valid inequalities that tighten the linear relaxation [42], [43]. Because the number of subproblems can grow exponentially with the number of integer variables, exact methods may become too slow for large instances, which motivates the use of metaheuristics such as Genetic Algorithm (GA) or Particle Swarm Optimization (PSO), which find good solutions quickly but without optimality guarantees [37]. In this thesis, the MILP models of Chapter 5 are solved with the Branch-and-Cut solver of the COIN-OR project through the PuLP modeling library (Section 4.5).

2.3 Human-Robot Collaboration

Even as factories increasingly adopt smart and connected technologies, human workers remain central to production. In this context, HRI constitutes the broadest concept, encompassing all forms of interaction between humans and robotic systems, ranging from fully separated operation to tightly coupled collaboration. Within this spectrum, HRC represents a subset of HRI in which humans and robots share a workspace and coordinate their actions to achieve a common goal. Building on this, HRTs correspond to a more structured and operational instantiation of HRC, where humans and robots are organized as interdependent agents with defined roles, shared objectives, and coordinated task execution.

HRTs, which embody the human-centric principles of Industry 5.0 [7], [8], leverage the complementarity between humans and robots to form synergistic partnerships. By combining human creativity, decision-making, and adaptability with the mechanical capabilities of collaborative robots, such as physical strength, precision, repeatability, and endurance [44], HRTs enable more flexible, efficient, and resilient manufacturing processes across multiple levels of interaction [8], [9].

This distinction is particularly relevant in the context of this thesis. Different forms of HRI imply different levels of spatial sharing, temporal coordination, and task interdependence, which directly affect both system-level scheduling and workstation-level interaction. While loosely

coupled interaction scenarios may require limited coordination, tightly integrated HRTs demand continuous synchronization between human and robot actions, as well as adaptation to system changes. Consequently, effective operation in such environments depends not only on optimized task allocation but also on intuitive and context-aware interfaces that support the operator in understanding and executing collaborative tasks.

According to Vicentini [45], seemingly self-explanatory terms such as collaboration, cooperation, and collaborative robot are often subject to inconsistent and misleading use in industrial and technical communication. Therefore, this thesis adopts the terminology proposed in [9].

- Collaborative robot/cobot: traditionally defined as a robot designed for direct interaction with a human within a defined collaborative workspace. Although this term remains widely used in literature and industry, International Organization for Standardization (ISO) 10218 now formally refers to *collaborative applications*, reinforcing that collaboration depends on the system configuration and risk assessment rather than on the robot itself [46].
- Collaborative application: an application that contains one or more collaborative tasks [46].
- Collaborative/shared workspace: a space within the safeguarded area where the robot and a human can perform tasks simultaneously during production operation [46]. In other words, it corresponds to the overlap between the human workspace and the robot's reachable workspace [47] apud. [48], as illustrated in Figure 2.2.

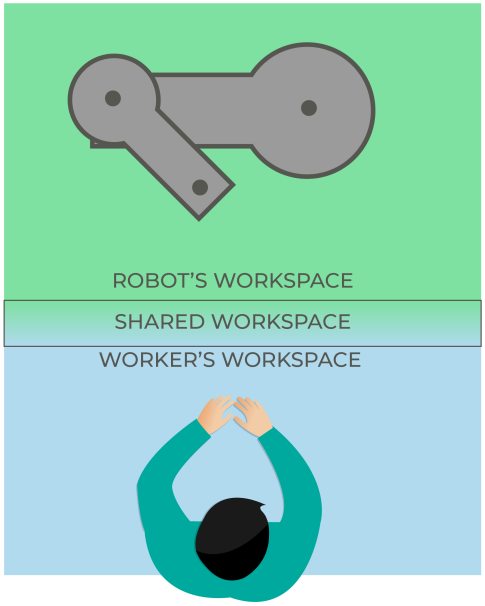


Figure 2.2: Exemplification of workers' and robots' workspaces, showing the overlapping areas where humans and robots can perform tasks simultaneously within a shared workspace.

- Human–Robot Interaction: a general term encompassing all forms of interaction between humans and robots [47] apud. [49]. It can be classified into five levels of interaction (cell, coexistence, synchronized, cooperation, collaboration), as shown in Figure 2.3.
- Cell: not a genuine cooperation scenario, since the human and the robot are separated by a safety cage [50].
- Coexistence: when the human operator and the robot are in the same environment but have separate workspaces [51].
- Synchronized: when the human operator and the robot work in the same shared workspace, but at different times [51].
- Cooperation: when the human operator and the robot work in the same shared workspace at the same time, although each performs a different task [51].
- Collaboration: when the human operator and the robot work in the same shared workspace, executing the same shared task simultaneously [46], [51].

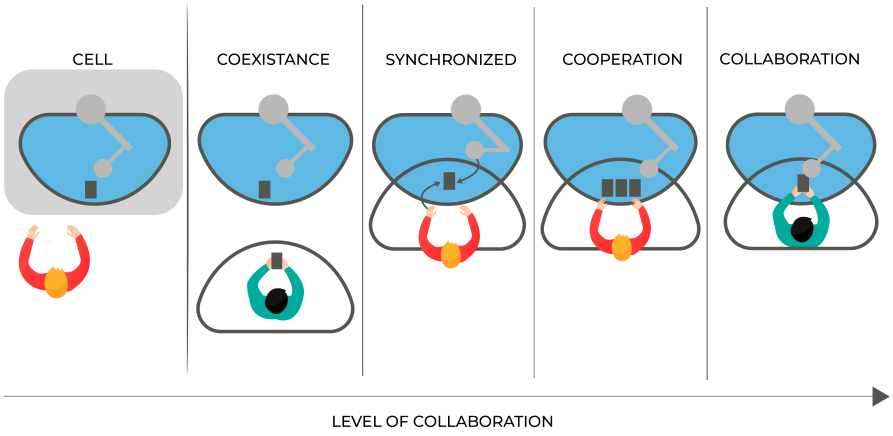


Figure 2.3: Levels of HRI, ranging from isolated work cells to full collaboration, illustrating the increasing degree of temporal and spatial coordination between humans and robots.

With the continuous growth in the adoption of industrial collaborative robots, the demand for seamless HRI has intensified. Effective cooperation between human operators and robotic agents is essential for achieving flexible, efficient, and high-performance production systems. Such collaboration can reduce task completion time and error rates, while simultaneously improving workplace ergonomics, lowering cognitive load, enabling decentralized decision-making, and optimizing the use of available resources [9].

In summary, effective HRC in industrial contexts demands precise coordination between human and robotic agents, clear definition of roles and responsibilities, awareness of system and

environmental states, and mechanisms to ensure operator safety. In addition, collaborative work requires support for decision-making and task execution, enabling humans to respond flexibly to dynamic conditions while maintaining productivity and quality. Meeting these requirements relies not only on the physical capabilities of robots but also on communication, feedback, and shared situational understanding between operators and machines.

AR has emerged as a promising tool to bridge these interaction gaps. By superimposing contextual digital information within the operator's field of view and enabling bidirectional input to CPPSs, AR helps ensure situational awareness, clarifies task assignments, and supports human decision-making, effectively addressing the "human-in-the-loop" integration challenge. Its effectiveness in HRI has been demonstrated across multiple application domains, including assembly guidance [52], [53], data visualization and feedback [54], safety enhancement [55], and robot programming and authoring [56], [57]. The growing market demand for collaborative robotics and AR technologies is reflected in an increasing number of research contributions focusing on AR-supported HRC [9].

2.4 Augmented Reality

In traditional assembly lines, operators often perform repetitive tasks on identical products, which limits the cognitive demand associated with task execution. However, in MMS and other flexible manufacturing environments, operators are frequently required to handle multiple products simultaneously, each with distinct structures, assembly sequences, and routing requirements. This variability, combined with dynamic task assignments and coordination with mobile platforms and robots, increases the cognitive and decision-making load on the operator [6], [58]. In such contexts, AR can provide substantial benefits by presenting on-demand context-sensitive instructions, visual cues, and system information, reducing errors and supporting timely decisions.

In industrial environments, assembly line operators are frequently required to handle diverse product structures and increasingly complex assembly sequences. This complexity, combined with the pressure to complete tasks within minimal time while maintaining high quality standards, makes optimal task execution challenging. In many cases, production systems lack a fully formalized task execution process, requiring operators to memorize extensive instructions and procedures for a wide range of products. Given the limited opportunities for repetition and consolidation of learning, this cognitive burden can hinder performance, leading to errors, rework, and reduced productivity [58].

To address these challenges, operators must be supported with tools that assist them during complex activities such as assembly of new products, quality assurance procedures, and maintenance operations. By superimposing digital information onto the physical workspace, AR can provide contextualized guidance, enhance understanding of manufacturing system behavior, and reduce reliance on memory-intensive instructions. This enables operators to execute assembly and maintenance tasks more efficiently and accurately, even in highly variable production environments [30].

According to Milgram and Kishino [59], AR refers to any scenario in which a real environment is enhanced with computer-generated objects. Milgram's reality-virtuality continuum has historically been influential in positioning AR between real and fully virtual environments, providing a conceptual framework for understanding mixed experiences, as illustrated in Figure 2.4. However, this model was developed in a technological context that predates many contemporary devices, interaction paradigms, and application domains. While valuable as a historical reference, it does not fully capture the diversity of modern Extended Reality (XR) implementations or current standardization efforts.

To address these limitations, contemporary reference models, such as the ISO/International Electrotechnical Commission 18039:2019 [60], have been developed. These models define architectural elements, functional components, services, and interaction features across real, augmented, and virtual environments, emphasizing context awareness, interoperability, and alignment with current hardware such as head-mounted displays and spatial computing devices. Within this framework, AR can be understood as a class of XR applications in which real-world sensory information is augmented with virtual content while maintaining the user's perception of the physical environment [61].

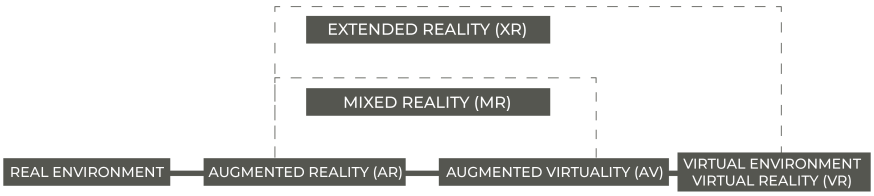


Figure 2.4: Overview of the XR continuum, illustrating the spectrum from fully real environments to fully virtual environments, including augmented, mixed, and virtual reality applications.

Azuma [62] further characterizes AR as any system that (i) combines real and virtual elements, (ii) enables interactive operation in real time, and (iii) ensures three-dimensional registration between virtual and physical objects. This definition highlights that AR is not restricted to specific hardware, and it reinforces the operational criteria that distinguish AR from purely virtual experiences.

As highlighted by Speicher, Hall, and Nebeling [63], the conceptual boundaries between immersive technologies such as Virtual Reality (VR), AR, Augmented Virtuality (AV), and Mixed Reality (MR) are often ambiguous, and no universally accepted definitions exist. Broadly, VR refers to technologies that fully immerse users in an artificial digital environment, isolating them from the physical world [64], whereas AR enhances the perception of the real environment by superimposing interactive digital information [9]. AV, in contrast, represents predominantly virtual

environments enriched with real-world elements or feedback [65], and MR provides a broader conceptual umbrella encompassing both AR and AV, where physical and virtual elements are blended to varying degrees [66], [67]. XR serves as an even more inclusive term, capturing the full spectrum of real-and-virtual combined environments and human-machine interactions enabled by immersive technologies [67], [68].

In the context of this thesis, these distinctions are not merely semantic. They help situate AR within the wider spectrum of immersive technologies and clarify its specific role in supporting operators in industrial environments. By enhancing situational awareness, providing context-sensitive guidance, and facilitating human-robot collaboration without isolating the user from the physical workspace, AR is positioned as the primary interface technology examined here. Recognizing the conceptual differences between AR, MR, and XR ensures that subsequent discussions of interface design, task support, and integration with scheduling algorithms are grounded in a clear, operationally relevant framework.

Although AR is still maturing for industrial applications due to hardware and software limitations, it has already demonstrated significant potential. Hardware challenges include ergonomic constraints, such as device weight and movement limitations, which restrict the integration of functionalities in wearable devices. Software limitations persist in tracking, recognition, and standardized development frameworks, along with unclear guidelines regarding which information to present and how [69], [70].

Despite these challenges, AR has been successfully applied in multiple industrial contexts, including enterprise resource planning [71] and Manufacturing Execution Systems (MESs) integration [15], assembly support [72], maintenance assistance [73], remote assistance [74], logistics [75], training [76], quality control [77], [78], data visualization [79], welding [80], product design [81], safety applications [82], and HRC, which will be discussed in Section 3.2.

Collectively, these applications demonstrate that AR can enhance operator performance, increase task efficiency, reduce errors and rework, and lower cognitive workload. For practical deployment, AR visualization technologies can be broadly categorized into four main types [69], [70], [83]–[85], each offering distinct advantages and constraints depending on the industrial task.

Head-Mounted Displays (HMDs), such as the HL2[1] and Magic Leap[2], provide optical see-through or video-see-through overlays of virtual content onto the user's field of view. Their primary strength lies in supporting hands-free operation and interaction in dynamic, mobile tasks-such as assembly, inspection, and HRC without forcing the operator to divert attention from the workspace. However, they present ergonomic challenges, such as, prolonged use can cause eye-strain and discomfort, and their weight and form factor may limit adoption in extended-shift environments or in tasks requiring high agility [84]. Additionally, HMDs may require calibration and may be sensitive to lighting conditions or occlusions .

Hand-Held Displays, including smartphones and tablets, employ video see-through AR, capturing the real environment via a camera and overlaying digital content. Their advantages include

[1] https://learn.microsoft.com/en-us/hololens/
[2] https://www.magicleap.com/magic-leap-2

low cost, easy deployment, and portability, making them suitable for inspection, training, and occasional assembly tasks. On the downside, they require manual handling, which can occupy both hands and potentially distract the operator from the primary task. They are therefore less suited for tasks that demand continuous hands-on operation or simultaneous interaction with multiple tools or components.

Static Screens, such as computer monitors or wall-mounted screens, can present augmented content without requiring wearable devices. They are well-suited for supervisory roles, monitoring, or tasks where operators can temporarily shift focus from their immediate work. Their main limitations include immobility, reliance on external cameras to capture the environment, the need for operators to redirect attention away from the workspace, and limited suitability for tasks that involve continuous movement or manipulation.

Spatial Displays, including projector-based systems, project augmented content directly onto real objects. This enables operators to maintain visual focus on the task while benefiting from contextual augmentation. Spatial AR is particularly useful for collaborative assembly, ergonomics guidance, or training scenarios where physical guidance cues are required. However, like fixed screens, spatial displays are stationary, and single-projector setups can suffer from occlusion, shadowing, or limited field coverage, restricting their use in complex or large-scale production areas.

In summary, each AR technology involves trade-offs between mobility, user immersion, hands-free operation, and cognitive load. HMDs emerge as particularly suitable for this thesis because they combine mobility with interaction and operator support, aligning with the requirements of HRC and dynamic task execution in matrix manufacturing environments. Other visualization technologies may still play a role in monitoring, training, or stationary tasks but are less appropriate for the operational scenarios considered here.

The use of HMDs has attracted increasing interest since 2018, largely due to their hands-free operation and enhanced mobility. The emergence of newer headsets capable of scene understanding and natural gesture interaction without external devices is expected to further accelerate research in this area [9], [18].

The precise placement of digital information is essential for ensuring a positive user experience in augmented reality systems and for achieving seamless integration between the physical and virtual environments [17]. Although there is no consensus on whether overlaid, adjacent, animated, or static-rendered Three-Dimensional (3D) models and instructions are most effective for different task types [86]–[88], misalignment of AR content relative to its physical counterpart can significantly impact user performance, depending on task complexity and nature. Consequently, computer vision-based tracking algorithms play a crucial role in mixed-reality systems, ensuring the accuracy, stability, and repeatability of displayed AR content [17].

The calibration of HMDs remains a critical challenge in medical and industrial applications, as it requires precise alignment between virtual and physical objects. To address this issue, some studies have proposed various online and offline calibration methods targeting the tracking device

[16], [89]–[91], while others have approached the problem from the perspective of markers [92], [93].

AR tracking systems can be broadly categorized into two main paradigms: outside-in (world-anchored) and inside-out (head-anchored), each with distinct operational characteristics, advantages, and limitations. Outside-in systems rely on fixed sensors in the environment to observe users and tracked objects. These setups typically offer high accuracy and reliability, making them suitable for precision-critical applications. However, they require external hardware that is often bulky and expensive, demand careful calibration, and are constrained by line-of-sight requirements. While effective in controlled environments, their flexibility and mobility in dynamic industrial settings can be limited [16], [88].

Inside-out systems embed sensors directly on the AR device, enabling self-contained tracking relative to the device's own coordinate frame. This approach provides high mobility, allows operation in unprepared environments, and aligns naturally with the user's perspective. Limitations arise from onboard computational and power constraints, which can reduce absolute accuracy compared to outside-in systems. Environmental factors such as lighting, texture, and rapid movement can further affect tracking performance [16], [88].

Within the inside-out paradigm, tracking can be further divided into marker-based, markerless, and hybrid approaches. Marker-based tracking uses fiducial markers to simplify localization and pose estimation, providing robustness and ease of implementation. However, it imposes workspace constraints, is sensitive to occlusion or dirt, and may introduce errors on the order of millimeters to centimeters. Markerless tracking leverages natural feature detection and Simultaneous Localization and Mapping (SLAM) algorithms, eliminating the need for artificial markers and improving usability in unprepared environments, though it requires higher computational resources and may reduce precision for small or fast-moving objects. Hybrid approaches combine multiple sensors-such as cameras, Inertial Measurement Unit (IMU), and depth sensors-to enhance robustness and tracking accuracy, particularly in dynamic scenarios, but increase system complexity and calibration effort [9], [94], [95].

In industrial HRC applications, selecting an appropriate tracking paradigm requires balancing these trade-offs. Tasks that demand high spatial precision, such as tool guidance or assembly verification, benefit from outside-in or hybrid tracking. Operators who move freely across multiple workstations are better served by inside-out, markerless, or hybrid approaches. No single tracking approach is universally optimal; understanding task requirements, environmental constraints, and cognitive demands is essential for designing effective AR-based support systems.

Spatial anchors provide a crucial mechanism to improve AR localization and coordination in shared environments. A spatial anchor is a fixed reference point in the physical workspace, typically established using marker-based tracking, that defines a known position and orientation in the world coordinate frame. These anchors enable correction of localization drift, facilitate consistent alignment of virtual content, and provide reliable reference points for computing spatial transformations between sensors, robotic systems, the physical environment, and the user's augmented view. In HRC, accurate spatial transformations are essential for visualizing robot trajectories,

coordinating tasks, and exchanging data between human and robotic agents [15], [96], [97].

Investigating the accuracy and precision of fiducial marker-based tracking in HMDs remains a central theme in current augmented reality research trends [98]. Such evaluations are essential to determine the viability of deploying these devices across diverse application domains, including medicine [99], architecture and civil engineering [100], [101], SLAM [102], and HRC [17], [55], [103], among others.

To successfully deploy Industrial Augmented Reality (IAR) systems in industrial contexts, it is essential to consider a combination of human, technical, and organizational factors, as well as temporal requirements that influence system performance. Human factors play a central role, user acceptance, engagement during testing, interface usability, ergonomics, and operator training are all critical to ensure that the system enhances efficiency without introducing frustration or cognitive overload. For instance, even technically robust AR systems may fail if operators find them uncomfortable or unintuitive to use [70], [83].

Technical factors include hardware and software considerations such as visibility and clarity of virtual information, voice control reliability, battery life, robustness of devices, and compatibility with existing Information Technology infrastructures. Scalability is also important, as industrial applications often require deployment across multiple workstations and users. In parallel, organizational factors, including management support, alignment with internal processes, and cost considerations, can significantly influence adoption and long-term sustainability.

From an implementation perspective, challenges arise in both technological and organizational domains. Limitations in software, processing speed, ergonomics, and system scalability can affect performance, while user acceptance, privacy concerns, and adaptation of workflows create additional hurdles. External factors such as integration with third-party systems, availability of technical support, and the absence of standardized industrial frameworks can further complicate deployment [104].

Finally, IAR systems must meet time-oriented requirements, ensuring that recurring processes like calibration or cleaning are rapid, the system maintains high reliability with minimal maintenance, and operations provide precise real-virtual alignment, tracking, and intuitive interaction. Additional considerations include minimizing motion sickness, improving operator ergonomics, and supporting safe human-robot interactions [70].Taken together, these factors highlight that effective IAR deployment is not only a matter of selecting the appropriate technology but also of integrating it thoughtfully into the human, operational, and organizational ecosystem.

According to Egger and Masood [83], Jetter, Eimecke, and Rese [105], and Bottani and Vignali [106], numerous scientific studies have sought to quantify the benefits of AR in industrial settings. Reported advantages include improvements in training quality and speed, enhanced operator safety, increased task execution efficiency, and overall performance gains. Additionally, AR has been associated with reductions in error rates during task execution, rework and redundant inspections, as well as decreased cognitive workload.

The discussion of AR technologies, tracking approaches, and operational considerations provides essential context for this thesis. In particular, understanding the capabilities and limitations

of different visualization and tracking methods is critical for designing interfaces that effectively support human-robot collaboration. This thesis focuses on comparing an AR-based interface with a traditional touch screen interface in a collaborative industrial scenario. By highlighting how AR can enhance the perception of robot actions, provide spatially aligned guidance, and reduce cognitive workload, the background presented here establishes why AR is a promising modality for this application. Moreover, awareness of practical constraints, such as tracking accuracy, ergonomics, and workspace integration, ensures that the comparison considers not only potential benefits but also the realistic challenges of deploying AR in collaborative environments.

Chapter 3

Related Works

The effective coordination of humans and robots in modern manufacturing systems has become increasingly important with the rise of flexible, modular production environments. Achieving high productivity and adaptability in such contexts requires not only sophisticated coordination of tasks and robotic resources but also interfaces that support operator awareness, intuitive decision-making, and seamless HRC. This thesis addresses both dimensions: it proposes a robotic coordination framework for HRT operating in a reconfigurable MMS, and evaluates two operator interfaces, an AR-based HMD and a conventional touch screen HMI, within that collaborative workstation context.

This chapter reviews the state of the art across these interrelated domains. Section 3.1 focuses on scheduling HRT within MMSs, beginning with FJSSP formulations and extending to approaches specifically addressing HRC, including team composition, role allocation, and mobile robot coordination. Section 3.2 examines AR interfaces in collaborative assembly scenarios, with particular attention to how different interface modalities influence operator awareness of robot behavior, perceived collaboration, and interaction quality. By analyzing these works, this chapter identifies the key challenges, methodological trends, and gaps in the literature, particularly the need for approaches that jointly consider task allocation, mobile robot coordination, and operator support through immersive interfaces. These insights, provided in Section 3.3, establish the foundation for the proposed methodology, which combines AR-enhanced interaction with optimization-driven scheduling to improve the performance of HRT in matrix manufacturing settings.

3.1 Scheduling Human-Robot Teams in Matrix Manufacturing Systems

To fully leverage the potential of MMS and HRTs, scheduling algorithms are required as decision-making tools to enhance production efficiency by assigning and sequencing operations to available resources while considering system constraints. The objective is to optimize one or more performance criteria by determining an optimal allocation of operations to resources ([6]). The most

frequently studied objectives are time-based (e.g., Makespan (C_{max}), completion time, tardiness, and earliness) and cost- or revenue-based (e.g., processing cost, logistics cost, machine utilization, and total production revenue). Each production system exhibits specific characteristics that can be mathematically modeled as constraints, most commonly involving precedence relationships among operations and resource availability ([24], [38]).

3.1.1 Flexible Job Shop Scheduling Problems

In the literature, the challenge of sequencing and allocating operations in arbitrary production environments with redundant workstations is widely recognized as the FJSSP. This problem has been extensively investigated as both a single-objective and a multi-objective optimization problem ([37], [107]).

Single-objective approaches span a spectrum from pure metaheuristics to hybrid formulations that combine exact MILP models with heuristic solvers, reflecting a trade-off between solution quality guarantees and computational tractability. For instance, Azzouz, Ennigrou, and Said [108] introduced a Self-Adaptive Hybrid Algorithm (SAHA) that combines a GA with iterated local search to minimize makespan while considering sequence-dependent setup times. Similarly, Fontes, Homayouni, and Gonçalves [109] developed a hybrid PSO and simulated annealing algorithm that incorporates transport resources into the scheduling process. In the context of sand-casting operations, Tang, Fang, Liu, *et al.* [110] proposed a Hybrid Teaching and Learning-Based Optimization Algorithm (HTLBOA) to improve scheduling efficiency. To address uncertainties such as processing-time variability and dynamic job insertion, Gao, Suganthan, Pan, *et al.* [111] presented a two-stage Artificial Bee Colony Optimization Algorithm (ABCOA), whereas Zhu and Zhou [112] introduced a Multi-Micro-Swarm Leadership Hierarchy-Based Optimization Algorithm (M2SLHO).

Extending classical formulations, Boyer, Vallikavungal, Cantú Rodríguez, *et al.* [113] compared MILP, Constraint Programming (CP), and Greedy Randomized Adaptive Search Procedure (GRASP) approaches for scheduling in rolled ring manufacturing, incorporating additional hard constraints including machine capacity limits, time lags, and sequence-dependent setup times. Zhao, Liu, Zhou, *et al.* [114] proposed a heuristic scheduling strategy integrating Petri Nets (PT nets) with an Iterated Greedy Algorithm (IGA) to minimize late jobs in wire rod and bar rolling processes. Shao, Shao, and Pi [115] investigated integrated lot sizing and scheduling in a distributed heterogeneous hybrid flow shop, introducing a Learning-Driven Iterated Local Search Algorithm (LDILS) enriched with Machine Learning (ML) techniques.

Multi-objective formulations extend the single-objective FJSSP by jointly optimizing makespan alongside additional criteria. Salido, Escamilla, Giret, *et al.* [116] proposed a GA to simultaneously optimize makespan and energy consumption, later extended by Salido, Escamilla, Barber, *et al.* [117] with a Memetic Algorithm (MA) for rescheduling disrupted operations. Li, He, Wang, *et al.* [118] employed Non-Dominated Sorting Genetic Algorithm II (NSGA II) to handle dynamic events while optimizing energy efficiency, and Dai, Tang, Giret, *et al.* [119] incorporated transportation constraints through an enhanced GA. Liu, Zhan, Chekem, *et al.* [120] proposed a

Hybrid Fruit Fly Optimization Algorithm (HFOA) considering carbon footprint reduction alongside makespan minimization. In distributed environments, Shao, Pi, and Shao [121] proposed a Pareto-based Estimation of Distribution Algorithm (PEDA) and Cheng, Zhang, Meng, *et al.* [122] employed a multi-threaded constraint programming assisted space perception evolutionary algorithm (MTCP-SPEA) for joint makespan and energy optimization. More recently, Chang, Tang, and Zhao [123] applied a Multi-Agent Reinforcement Learning-enhanced Lagrangian Decomposition (MRLLD) in steelmaking and Ben Hamida, Azzouz, and Ben Said [124] proposed a hybrid Self-learning Variable Neighborhood Search (SL-VNS) with Q-learning for dynamic parameter tuning.

Comprehensive reviews of the FJSSP are provided by Destouet, Tlahig, Bettayeb, *et al.* [24], Dauzère-Pérès, Ding, Shen, *et al.* [37], Xiong, Shi, Ren, *et al.* [38], and Türkyılmaz, Şenvar, Ünal, *et al.* [107]. While these works demonstrate strong optimization performance, they focus almost exclusively on benchmark instances and rarely address deployment in physical manufacturing systems, HRC, or mobile robot coordination. A summary of the reviewed works is presented in Table 3.1[3]. The present work draws on FJSSP concepts for operation sequencing but extends them toward the physical coordination of HRT in a reconfigurable matrix manufacturing environment.

3.1.2 Human-Robot Teams Scheduling Problems

While the traditional FJSSP offers a well-established framework for allocating tasks to machines in highly automated environments, it is inherently designed for independent and predictable resources and fails to capture the unique challenges posed by HRC. Scheduling for HRT introduces additional layers of complexity, as tasks must be allocated across heterogeneous agents with fundamentally different capabilities. Human operators exhibit variability in skills, adaptability, and performance influenced by factors such as fatigue or learning, whereas robots provide consistency but limited flexibility. Furthermore, many tasks require simultaneous and coordinated execution, creating temporal and logical interdependencies that traditional formulations do not address. These challenges underscore the need for dedicated models tailored to HRT environments ([7], [20], [125]).

In scheduling literature, the allocation, sequencing, and assignment of HRT modes to manufacturing operations is typically formalized as the Multi-Mode Task Scheduling Problem (MMTSP), a generalization of the Parallel Machine Scheduling Problem. The term *multimode* reflects the ability of HRTs to perform operations in three distinct modes, human-only, robot-only, or HRC. The term *multiprocessor* emphasizes operations that require simultaneous coordination between human and robotic agents ([7], [40]). The literature review by Christoph Petzoldt and Freitag [23] highlights an increasing emphasis on dynamic and adaptive task allocation strategies for HRTs in assembly operations, with optimization algorithms such as GA, B&B, and Ant Colony Optimization (ACO) commonly used to improve allocation efficiency while adhering to constraints including resource characteristics, cycle time, ergonomics, workload, and safety.

[3]The symbol "–" indicates that the corresponding work did not provide a mathematical formulation.

Table 3.1: Overview of key studies addressing the FJSSP, including their problem formulation, solution approach, primary optimization objectives, and target application domains.

Study	Formulation	Approach	Objectives	Application
[108]	–	SAHA	C_{max}	Benchmark
[109]	MILP Lower bound	PSO Simulated annealing	C_{max} Exit time	Benchmark
[110]	MILP Disjunctive graph	HTLBOA	C_{max}	Benchmark Sand Casting
[111]	–	ABCOA	C_{max}	Benchmark
[112]	MILP	M2SLHO	C_{max}	Bicycle Production
[113]	MILP CP Disjunctive Graph	MILP CP GRASP	C_{max}	Ring Rolling
[114]	PT net MILP	IGA	# of late Jobs	Benchmark
[115]	MILP	LDILS	Total Tardiness	Benchmark
[116]	–	GA	C_{max} Energy Consumption	Benchmark
[117]	–	MA	C_{max} Energy Consumption	Benchmark
[118]	–	NSGA II	C_{max} Energy Consumption	Benchmark
[119]	MILP	GA	C_{max} Energy Consumption	Benchmark
[120]	MILP	HFOA	C_{max} Carbon Footprint	Benchmark
[121]	MILP	PEDA	C_{max} Total Weighted Tardiness	Benchmark
[122]	CP	MTCP-SPEA	C_{max} Energy Consumption	Benchmark
[123]	MILP	MRLLD	C_{max} Energy Consumption	Steelmaking
[124]	–	SL-VNS	C_{max} Energy Consumption	Benchmark

A first line of work frames HRT task allocation as a combinatorial optimization problem. Building on Bianco, Dell'Olmo, and Grazia Speranza [39] and Bianco, Dell'Olmo, Giordani, *et al.* [40] as a benchmark, Ferreira, Figueira, and Amorim [7] addresses the MMTSP with a novel MILP formulation and two solution methods, CP and GA, both outperforming the benchmark. Raatz, Blankemeyer, Recker, *et al.* [126] presents a GA task scheduling approach for HRC workcells accounting for human and robot execution times and operational factors such as weight, material rigidity, and ergonomic considerations, validated at Lenze SE through gearbox assembly optimization. N. Gjeldum and Mladineo [127] employs ACO to further improve efficiency and flexibility in HRC tasks in a similar gearbox assembly context. Addressing HRC disassembly processes, Liao, Chen, Hu, *et al.* [128] proposed an optimization framework integrating disassembly cost, operator safety, and task complexity into a multi-attribute utility function; this work

was extended and experimentally validated by "Human-robot collaboration for e-waste disassembly: task modeling, allocation, and ergonomic assessment" [129] with 22 participants, showing reduced workload and improved ergonomics despite increased task completion time.

A second line shifts focus toward robustness under uncertainty, addressing temporal variability in human execution through dynamic and reactive scheduling strategies. Tchane Djogdom, Meziane, and Otis [20] reformulated the HRT scheduling problem as a combination of the Travelling Salesman Problem and a binary Knapsack Problem, modeled through an AND/OR Graph (AOG) and solved using Dynamic Programming, with reactive rescheduling validated through simulation. Lamon, De Franco, Peternel, *et al.* [130] and Merlo, Lamon, Fusaro, *et al.* [125] adopt AOG-based scheduling strategies for ergonomic optimization in HRC assembly of metallic structures, with the latter integrating motion capture systems to dynamically update scheduling parameters based on ergonomic risk estimates. Beyond graph-based formulations, Casalino, Zanchettin, Piroddi, *et al.* [131] proposed a scheduling framework based on Time PT net validated in a physical assembly station with two robots and one human operator, and Yu, Lu, Yin, *et al.* [132] introduced a Self-Learning Memetic Algorithm (SLMA) minimizing both makespan and total energy consumption in a distributed mixed fuzzy welding shop. A summary of the reviewed works is provided in Table 3.2[3].

Table 3.2: Overview of key studies addressing HRT scheduling problems, detailing problem formulations, solution approaches, optimization objectives, application domains, and implementation status.

Study	Formulation	Approach	Objectives	Application	Implemented
[7]	MILP Lower bound CP	CP GA	Min. C_{max}	Benchmark	No
[126]	–	GA	Min. cycle time Max. ergonomics	Gearbox assembly	Yes
[127]	–	ACO	Min. cycle time Min. investment cost Min. workspace layout Min. worker effort	Gearbox assembly	Yes
[128]	AOG MILP	Exact (enumeration)	Max. overall utility	Computer disassembly	No
[129]	AOG MILP Hierarchical task analysis	Exact (enumeration)	Max. overall utility	Computer disassembly	Yes
[20]	AOG	Dynamic programming	Min. resource idle time	Extraction of battery bells	Simulation
[130]	AOG	A*	Max. ergonomics	Assembly of metallic profiles	Yes
[125]	AOG	AO*	Max. ergonomics	Assembly of metallic profiles	Yes
[131]	Time PT net	B&B	Min. agent idle time	Assembly of USB/microSD adapter	Yes
[132]	MILP	SLMA	Min. C_{max} Min. energy consumption	Benchmark	No

While prior studies have addressed isolated aspects of HRT scheduling, none have tackled the combined coordination of HRT, mobile robot transportation, and battery management within a MMS. Works focused on FJSSP generally overlook HRC, while HRT scheduling studies typically assume fixed layouts and do not incorporate mobile robot logistics. The framework proposed in this thesis bridges this gap by embedding coordination logic for heterogeneous human and robotic agents directly into a deployed industrial MMS, where operation assignment, robot relocation, and battery swapping are handled jointly through a coordinated decision pipeline, and the approach is validated through a proof-of-concept implementation, directly addressing the persistent gap between theoretical formulation and practical deployment.

3.2 Augmented Reality in Human-Robot Collaborative Assembly Scenarios

The application of AR to enhance HRC in assembly environments has attracted growing research interest, driven by the technology's ability to overlay contextual information directly onto the physical workspace, thereby supporting operator guidance, situational awareness, and intuitive interaction with robotic systems. In shared workspaces, where operators must coordinate with robotic agents, the choice of interface modality directly influences how well operators understand robot behavior, anticipate upcoming actions, and respond to collaborative task demands. This review focuses specifically on HMD-based approaches, as they represent the most immersive and industrially relevant form of AR interaction for HRC scenarios, with all selected studies relying on either the HL1 or HL2 as the visualization device.

Beyond instruction delivery, AR has been used in HRC for multi-user interaction, robot programming, and intent communication. Leveraging marker-based spatial anchors, Martins, Costa, Petry, *et al.* [133] developed an AR application that enables multiple users to simultaneously visualize and interact with several robot Digital Twins (DT) within a shared workspace. Similarly, Soares, Petry, and Moreira [56] and Blankemeyer, Wiemann, Posniak, *et al.* [134] implemented programming-by-demonstration approaches supported by AR, allowing operators to intuitively guide robot behavior. Kalpagam Ganesan, Rathore, Ross, *et al.* [53] projected intended robot motions and task instructions to help operators anticipate robot actions, thereby improving safety and coordination during manipulator interaction. More recently, Santos [135] presented a system for a collaborative manipulator that integrates dynamic collision avoidance, holographic previews of robot trajectories, and an interactive user interface, further illustrating the potential of AR to enhance HRC in operational contexts.

3.2.1 Interface Design and Operator Awareness in Human-Robot Collaboration

A recurring finding across studies is that AR interfaces improve operators' spatial understanding of robot actions and task context compared to conventional screen-based alternatives. Hietanen,

Pieters, Lanz, *et al.* [55] evaluated AR interfaces during diesel engine assembly, finding that AR-supported conditions reduced task time by 21–24% and robot idle time by 57–64% compared to a non-collaborative baseline; notably, users favored a projector-based system over the HMD in usability, illustrating that modality effects on interaction quality are not straightforwardly determined by performance alone. Ajikumar, Marsella, and Moghaddam [136] developed an AR system using the HL2 for gearbox assembly, finding that visual feedback was a critical factor. Its absence reduced user confidence and trust in verbal commands, while its inclusion improved task completion times, reduced errors, and increased overall trust, with National Aeronautics and Space Administration - Task Load Index (NASA-TLX) results confirming significantly reduced mental workload.

The role of spatial visualization in supporting robot intent communication has also been explored. Chu and Liu [19] examined two AR-based interfaces alongside haptic feedback configurations, finding that a visual interface projecting the gripper's future position seconds ahead of its actual movement notably reduced users' gaze time toward the robot, suggesting that interface-mediated anticipation of robot motion can reduce the monitoring burden on the operator. Participants favored this gripper trajectory visualization in both usability and perceived safety. In the context of Physical Human-Robot Interaction (pHRI), Madani, Niaz, Guler, *et al.* [137] proposed an AR system integrated with adaptive admittance control for curved-surface drilling tasks, finding that combining AR with haptic guidance improved task performance by 26%, reduced perceived effort by 16%, and lowered muscle activation by 27%.

Studies have also examined how interaction modality affects workload and usability. Tadeja, Zhou, Capponi, *et al.* [138] compared eye-gaze and hand-ray pointing methods for part selection, finding that eye-gaze enabled faster task completion while hand-ray was associated with slightly higher mental workload, with both approaches achieving mean System Usability Scale (SUS) scores around 80%. Sanna, Manuri, Fiorenza, *et al.* [139] proposed an AR-Brain-Computer Interface integrating the NextMind[4] neural interface with the HL2, achieving a SUS score of 76%, though with higher mental, temporal and effort demands. Further evidence of AR's benefits is reported by Dimitropoulos, Togias, Zacharaki, *et al.* [140] (6% cycle time reduction and 80% ergonomics improvement in elevator cab door assembly), Tong, Zhang, Galeano, *et al.* [141] (gains in task effectiveness and user experience in custom part assembly), Apostolopoulos, Andronas, Fourtakas, *et al.* [142] (SUS of 82% in powertrain assembly), Andronas, Apostolopoulos, Fourtakas, *et al.* [143] (SUS of 78% in powertrain assembly with multimodal interaction), and Dalle Mura and Dini [144] (significant assembly time reduction in car panel fitting). More recently, Tadesse, Wang, Lin, *et al.* [145] proposed a YOLO-based object detection-driven AR display achieving faster cycle times and reduced workload, and Xu, Zhou, Zhou, *et al.* [146] proposed an AR–Large Language Model (LLM)–Vision Language Model (VLM)–robot framework demonstrating advantages in task success rate and reasoning latency, though this work remains a preprint and has not yet undergone peer review. Table 3.3 summarizes the reviewed works.

[4] https://github.com/Snapchat/NextMind

The studies reviewed in this section indicate that AR-based interfaces can improve operator performance, reduce cognitive and physical workload, and enhance situational awareness in collaborative assembly tasks. However, most studies evaluate AR against no-interface or paper-based baselines, or compare different AR interaction techniques against one another. Fewer works conduct a direct, controlled comparison between an AR interface and a conventional industrial HMI under matched task conditions. As a result, it remains difficult to isolate the effect of interface modality itself on human-robot interaction quality, including operator awareness of robot behavior, perceived collaboration, workload, and usability. The interface comparison proposed in this thesis addresses this gap directly.

Table 3.3: Overview of selected AR-based HRC studies without scheduling algorithms, detailing task types, robotic platforms, AR devices, alignment methods, quantitative and qualitative evaluation methods, and key HRI outcomes.

Reference	**Task**	**Robot**	**Devices**	**Alignment**	**Quantitative**	**Qualitative**	**Key HRI Outcomes**
[140]	Elevator cab door assembly	COMAU AURA	HL2	Unidentified fiducial markers	Execution Time; Rapid Upper-limb Assessment (RULA)	Custom usability questionnaire	6% cycle time reduction; 80% ergonomics improvement; no interaction quality measures
[141]	Custom part assembly	KUKA LBR iiwa	HL2	–	–	Usability Metric for User Experience	Improved task effectiveness, efficiency, and user satisfaction
[55]	Engine assembly	Universal Robots (UR)5	HL1; Projector	Unknown fiducial marker	Execution time; Robot idle time	Custom User Experience (UX) questionnaire	AR reduced robot idle time by 57–64%; users preferred projector over HMD in usability
[144]	Panel fitting	–	HL2	Quick Response (QR) Code	Execution time	–	Assembly time reduced
[142]	Turbocharger powertrain assembly	UR5	HL2	–	–	SUS	SUS of 82%
[143]	Powertrain assembly	UR5	Tablet; Smartwatch; HL2	–	–	SUS	SUS of 78%; multimodal interaction accepted

continued on next page

Table 3.3 – *continued from previous page*

Reference	Task	Robot	Devices	Alignment	Quantitative	Qualitative	Key HRI Outcomes
[136]	Gearbox assembly	Sawyer	HL2	–	Execution time; Error rate	NASA-TLX; Custom trust questionnaire	AR visual feedback improved trust and reduced mental workload; absence reduced operator confidence
[138]	Control station assembly	Franka Emika Research	HL2	QR Code	Execution Time	SUS; NASA-TLX; Simulation Sickness Questionnaire (SSQ)	Eye-gaze faster with lower workload than hand-ray; users comfortable working alongside robot
[139]	Vise assembly	COMAU e.Do	HL2	–	Execution Time	SUS; NASA-TLX	SUS of 76%; higher mental and effort demands; low physical demand and frustration
[19]	LEGO assembly	WidowX 250	HL2; SenseGlove Nova	QR Code	Execution Time; Hand proximity; Gaze time	SUS; Safety questionnaire	Gripper trajectory preview reduced gaze time toward robot; improved perceived safety
[137]	Drilling curved surfaces	KUKA LBR iiwa 7	HL2	Unidentified fiducial marker	Execution Time; Human effort; Muscle activation; Alignment error	Custom usability questionnaire	AR and haptic guidance improved performance by 26%; reduced effort and muscle activation
[145]	GPU assembly	Techman	HL2	–	Execution Time	NASA-TLX	Faster cycle times and reduced mental workload

continued on next page

Table 3.3 – *continued from previous page*

Reference	Task	Robot	Devices	Alignment	Quantitative	Qualitative	Key HRI Outcomes
[146]	Custom part assembly	Rokaw ER3	HL2	–	Task success rate; Response time	NASA-TLX	Superior task success rate and reasoning latency over baselines (preprint)

3.2.2 Interface Modality in Scheduled and Dynamic Human-Robot Collaboration

A more recent line of work has examined interface design in contexts where task allocation between human and robot agents is managed algorithmically, introducing additional demands on the operator to interpret system state, understand task ownership, and coordinate actions under dynamic scheduling conditions. These scenarios place greater emphasis on the interface's role as a coordination support tool rather than merely an instruction display.

Within this literature, a first methodological line relies on classical optimization and search strategies. De Franco, Lamon, Balatti, *et al.* [54] investigated AR in a collaborative surface-polishing task coordinated through a task allocation algorithm, finding that participants strongly preferred the AR interface over conventional screen-based feedback, citing enhanced focus and clearer task comprehension. The task allocation strategy from that work was detailed and further evaluated by Lamon, De Franco, Peternel, *et al.* [130] in a collaborative metallic structure assembly context, showing that the AR empowered HRC condition reduced both physical and cognitive effort compared to a manual baseline. Building on this, Lamon, Fusaro, De Momi, *et al.* [25] refined the framework by introducing a negotiation phase and replacing the AOG and A* strategy with BT combined with a MILP model; the AR empowered HRC configuration produced the fastest task completion times and reduced perceived workload across all negative NASA-TLX subscales.

A related strand shifts the locus of scheduling decisions toward the operator. Lunding, Lystbæk, Feuchtner, *et al.* [26] proposed a framework where the task sequence is predefined but the AR interface enables operators to decide in real time which agent should execute a task when both are eligible. This flexibility led to more dynamic forms of collaboration, with participants more inclined to work in parallel with the robot. Hologram misalignment was identified as a potential source of errors, and participants emphasized the importance of clearly visualizing task ownership, a finding directly relevant to the role of interface design in supporting coordination. Higher eye fatigue was observed in AR-based conditions compared to alternative interfaces.

A second methodological line replaces combinatorial optimization with reinforcement learning. Liu, Tang, and Zhang [147] applied a MA-A2C Reinforcement Learning (RL) algorithm for planning and scheduling in turbofan engine assembly, extended in Liu, Tang, Zhu, *et al.* [148] with the integration of a VLM; however, no user evaluation was conducted in either paper. In contrast,

Table 3.4: Overview of selected AR-based HRC studies in scheduled collaboration contexts, detailing task types, robotic platforms, AR devices, algorithmic approaches, quantitative and qualitative evaluation methods, and key HRI outcomes.

Reference	Task	Robot	Devices	Algorithm	Quantitative	Qualitative	Key HRI Outcomes
[54]	Surface polishing	Franka Emika Panda on Robotnik Summit-XL Steel	HL1	AOG; A*	Applied Force	Custom usability questionnaire	AR strongly preferred over screen; improved task focus and comprehension
[130]	Metallic structure assembly	Franka Emika Panda	HL1	AOG; A*	–	Custom usability questionnaire	HRC and AR reduced physical and cognitive effort; instructions perceived as intuitive
[25]	Metallic structure assembly	Franka Emika Panda	HL2	Behavior Trees (BT); MILP	Execution time	NASA-TLX; Custom questionnaire	HRC and AR produced fastest completion; workload reduced across negative NASA-TLX subscales
[26]	LEGO assembly	UR5e	HL2	Custom user-mediated scheduling	Execution Time	NASA-TLX; Custom questionnaire; Semi-structured interview	AR users more likely to work in parallel; task ownership visibility critical; higher eye fatigue in AR
[147]	Turbofan engine assembly	FR5	HL2	Multi-Agent Advantage Actor-Critic (MA-A2C)	–	–	No user evaluation conducted
[148]	Turbofan engine assembly	FR5	HL2	MA-A2C; VLM	–	–	No user evaluation conducted
[149]	automated guided vehicle (AGV) steering wheel assembly	Aubo i5	HL2	Multi-Agent Transfer Reinforcement Learning (MA-TRL)	Execution time	–	RL integration improved task performance; AR used for operator guidance and robot training
[150]	Timber frame assembly	ABB IRB 6400	HL2	Custom decision tree	–	–	No user studies conducted

Wang, Yan, Hu, *et al.* [149] implemented and experimentally evaluated a MA-TRL scheduling strategy for collaborative assembly of an AGV steering wheel, where the RL model was trained through an AR interface that also provided real-time assembly guidance, showing that integrating RL significantly improved task performance. Outside these two main lines, Rogeau and Rezaei Rad [150] proposed a task scheduling strategy for AR-assisted timber frame assembly based on physical and structural parameters, though without user-based validation. A summary of the reviewed works is presented in Table 3.4.

3.3 Conclusion and Research Gaps

Although extensive research has been conducted on both FJSSP and HRT scheduling, these domains have largely evolved in parallel. FJSSP formulations focus on optimizing machine allocation and task sequencing, typically overlooking human-related factors and collaborative operating modes, and are validated almost exclusively on benchmark instances. HRT scheduling approaches explicitly address human-robot interaction constraints but assume simplified production environments, typically fixed layouts or single workstations, and lack the structural complexity of FJSSP models. Consequently, the system-level logistics layer that emerges in real MMS deployments, where routing, relocation, and battery management of mobile robots must be coordinated with task allocation across multiple HRT stations, is absent from all reviewed works.

This thesis addresses this gap by proposing a robotic coordination framework that embeds coordination logic for heterogeneous human and robotic agents directly into a deployed industrial MMS, where operation assignment, robot relocation, and battery swapping are handled jointly through a coordinated MILP-based decision pipeline. Unlike prior works that treat these as separate problems, the proposed framework coordinates them within a single sequential structure and validates the approach through a proof-of-concept implementation on a industrial-deployed MMS platform, directly addressing the persistent gap between theoretical formulation and practical deployment.

As highlighted in the literature review, AR has demonstrated strong potential to enhance HRC at the workstation level. However, most studies evaluate AR as a standalone guidance layer, independently of any production planning mechanism, and compare it against no-interface or paper-based baselines rather than against a methodologically equivalent conventional industrial HMI. When operators work within a coordinated MMS governed by optimization-driven decisions, the interface must do more than deliver assembly instructions, it must convey task-specific guidance consistent with system-level scheduling outputs. Moreover, the question of how interface modality itself affects human-robot interaction quality, operator awareness of robot behavior, perceived collaboration, workload, usability, and execution performance, cannot be answered without a controlled comparison under matched task conditions. A direct, controlled comparison between an AR interface and a conventional industrial HMI under identical task conditions and instructional logic, enabling observed differences to be attributed to interface modality rather than content, is absent from all prior work in the reviewed literature.

This thesis addresses this gap by embedding the AR interface within a scheduling-driven production context and conducting a controlled comparative evaluation between the AR interface and a conventional industrial touch screen HMI, using identical instructional logic across both modalities. This enables a rigorous assessment of how interface modality affects operator performance, workload, and the perceived quality of collaboration with the robot in a shared industrial workspace.

Chapter 4

System Architecture Description

This chapter presents the system architecture underpinning the proposed framework, establishing the technical foundation necessary for understanding the subsequent methodological contributions. However, the purpose of this chapter is not merely to catalog the components that were assembled, but to articulate why each architectural decision was made, what problem it solves, and how it contributes to answering the central research questions of this thesis. The architecture described here is the integrative substrate that makes possible the end-to-end pipeline of the thesis, from MILP-based schedule optimization, through physical task execution on the Prolynk MMS, to the contextual delivery of operator instructions via AR. Each design choice reflects a deliberate response to the constraints and requirements imposed by flexible HRC in industrial environments, where the coordination between human operators, robotic agents, and scheduling logic must be seamless, reliable, and traceable.

Before detailing individual elements, an overview of the system as a whole is provided, explaining how the MMS, the optimization algorithm, the tracking system, the communication framework, and the operator interfaces are interconnected and collectively support the proposed approach. Building on this integrated perspective, the chapter then presents a detailed description of the hardware components, software modules, and communication protocols, with emphasis not only on the technologies selected but also on the rationale behind each design decision and its contribution to efficient HRC, reliable data processing, and AR integration. By the end of this chapter, the reader will have a clear understanding of both the architecture as a unified system and the infrastructure that enables the coordinated operation of HRT across the experimental scenarios explored in later chapters.

4.1 System Overview

A central challenge in designing a framework for AR-supported HRC in flexible manufacturing is bridging the gap between high-level scheduling decisions and low-level physical execution, while simultaneously ensuring that human operators receive the right information at the right time in a form that is both contextually meaningful and spatially coherent. The architecture proposed

in this thesis addresses this challenge by integrating five functionally distinct but tightly coupled elements: the Prolynk MMS, the MILP-based optimization framework, the marker-based tracking system, the communication framework, and the operator interfaces.

The Prolynk MMS constitutes the physical substrate of the system, comprising a set of workstations staffed by HRTs and a fleet of MPCs responsible for transporting workpieces between stations. Coordinating this environment is the MILP optimization framework, which runs on a dedicated server and generates task schedules that account for operation assignments, transportation routes, and battery management constraints. Once a schedule is produced, its decisions are propagated to the physical layer. MPCs are dispatched along predefined routes, and the UR5 manipulator at the collaborative station receives its operation sequences via the Task Manager. This tight coupling between the optimization layer and the execution layer is what allows the system to function as a coherent whole rather than a collection of independent subsystems.

Simultaneously, the computed schedule drives the AR instruction delivery pipeline. Each operator is equipped with a HL2 HMD, which renders task-specific guidance directly within their field of view. The choice to deliver instructions in this way, rather than through a conventional fixed-screen HMI, is deliberate since it preserves operator attention and mobility, reduces the cognitive load associated with context-switching between a display and the physical task, and allows instructions to be spatially anchored to the relevant workspace elements. Stable registration of virtual content onto the physical environment is achieved through native QR code tracking, enabling reliable spatial anchoring as operators move freely across the shop floor.

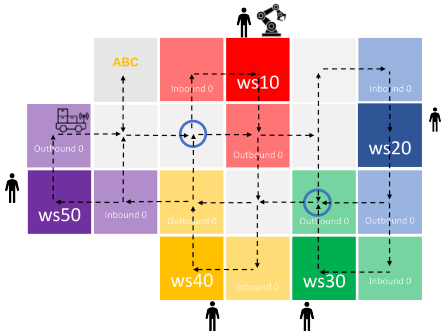
Underpinning this end-to-end flow is a communication framework that connects the coordination service, the robotic resources, the MMS controller, the operator interfaces, and a data repository. It is described in Section 4.5 in terms of functions and information flows, independently of specific products, while the technologies used in this implementation are listed separately. The following sections describe each component in detail and the rationale behind the design decisions.

4.2 Matrix Manufacturing System

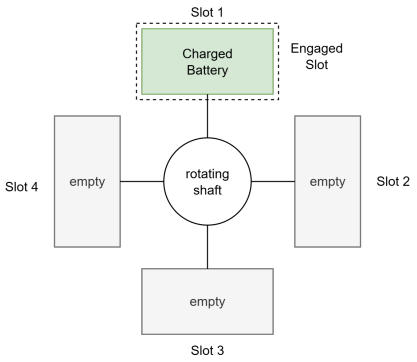
The MMS considered in this study comprises five Workstations (WSs) (WS 10–WS 50) and one Automatic Battery Changer (ABC), as depicted in Fig. 4.1a. The ABC, shown schematically in Fig. 4.1b, consists of a rotating shaft and four battery slots that enable the system to retrieve a depleted battery from a mobile robot and insert a charged one during the same visit; this slot-based exchange mechanism is the operational model underlying the Battery Swap module.

Of the five workstations, WS 20 and WS 50 were excluded from the experimental use case. Because transportation routes in the Prolynk system are fixed, these two stations introduce path-blocking risks that the scheduling models proposed in Chapter 5 do not account for. WS 20 occupies the only route between WS 10 and WS 30, and WS 50 blocks access between the remaining workstations and the ABC. Activating either station would therefore risk obstructing material flow in ways that fall outside the assumptions of the scheduling formulation. Their exclusion is

a methodologically motivated decision that ensures the experimental conditions remain consistent with those assumptions.



(a) Layout of the material flow routes within the Prolynk MMS, illustrating the possible paths followed by mobile carriers between workstations.



(b) Simplified representation of the Prolynk ABC, highlighting its structural components and role in enabling modular transport and positioning within the system.

Figure 4.1: Overview of the Prolynk MMS, showing the routing flexibility of the system and the underlying transport platform that supports dynamic product flow between workstations.

Among the active stations, WS 30 and WS 40 are manual workstations, whereas WS 10 operates as a collaborative station equipped with a UR5 manipulator and a Robotiq 2F-140 gripper[5]. Each workstation is equipped with a touch screen HMI for operator interaction, consisting of an industrial mini-PC and a capacitive touchscreen display (specifications in Table 4.6). These HMIs serve a dual purpose in the architecture, they are the primary interface for operators who are not equipped with the HL2 HMD, and they also provide the baseline condition against which the AR interface is experimentally compared to.

Product transportation between stations and the ABC is carried out by a fleet of three MPCs, following the allocation decisions generated by the scheduling models and predefined trajectories. The MPCs are compact mobile platforms measuring 450 x 350 x 100 mm, equipped with four compliant balancing wheels and driven by two encoder wheels. They can reach speeds of up to 400 mm/s while carrying a maximum payload of 15 kg. Powered by a 24 VDC battery, each MPC achieves an autonomy of up to 8 hours of uninterrupted operation. Each platform is fitted with a 450 x 385 x 8 mm product tool plate, which is pneumatically actuated against indexing pins to ensure surface stability during operations. The plate also integrates an I/O module, enabling the development of smart tooling with embedded sensors and/or actuators managed by the PLC, even while the MPC is in motion. From the perspective of the scheduling framework, the MPCs are not merely transport devices but constrained resources whose availability, routing, and energy state must be jointly optimized alongside human and robotic operations. This is why the scheduling models developed in this thesis explicitly account for transportation routes and battery management.

[5] https://robotiq.com/products/adaptive-grippers

In this system, empty slots retrieve depleted batteries from the MPCs, while occupied slots insert fully charged batteries back into them. To ensure reliable operation, a battery swap threshold of 20% was established based on experimentally obtained charge and discharge curves under continuous operation. This threshold guarantees that each robot can safely complete a battery replacement cycle before full depletion, even when all three robots operate concurrently with only one spare battery. The charge and discharge curves presented in Figure 4.2 and Figure 4.3 were logged using the RIGOL DC Load DL3031[6] and the MPC onboard battery-level sensor, respectively.

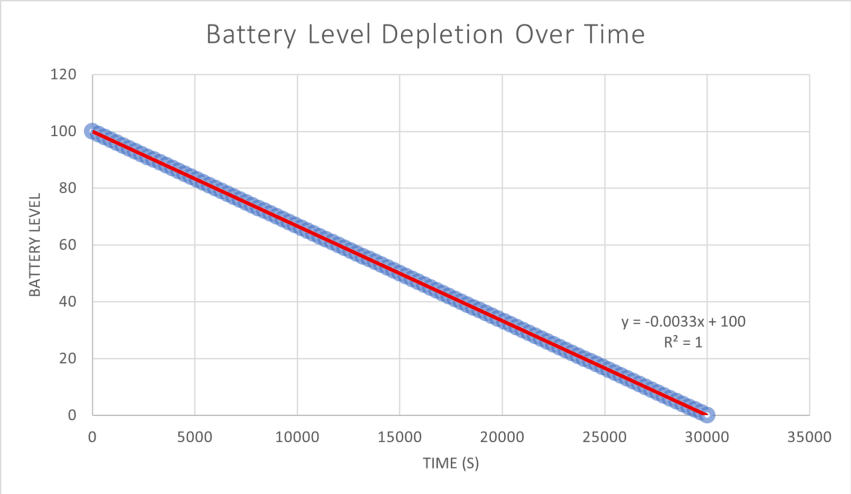


Figure 4.2: Evolution of the battery charge level of the MPCs over time during operation, illustrating their discharge behavior under typical task execution conditions.

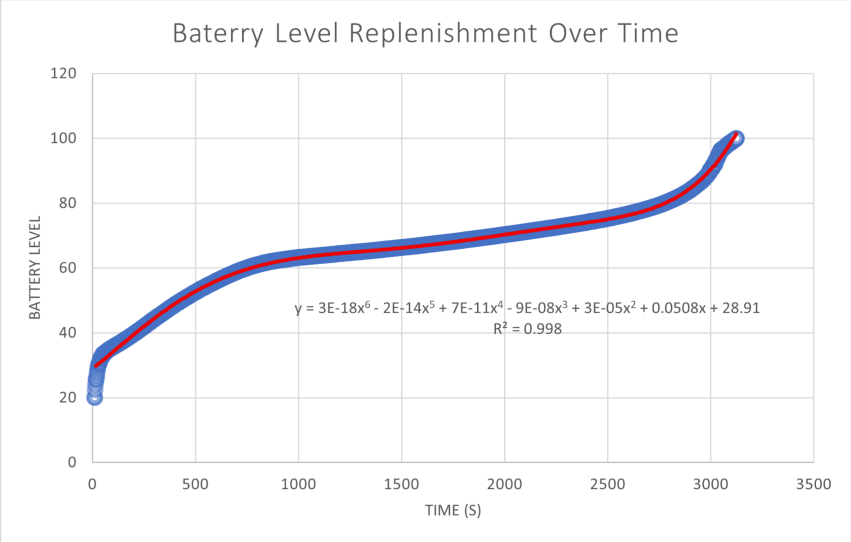


Figure 4.3: Evolution of the battery charge level of the MPCs during charging cycles, showing their replenishment dynamics and recovery profile.

[6] https://rigolshop.eu/products/dc-load/dl3031.html

4.3 Head-Mounted Display

Within the proposed architecture, the HL2 serves as the operator-facing output layer of the system, through which the decisions produced by the MILP optimization framework are translated into contextual, task-specific guidance delivered directly within the operator's field of view. Rather than relying on a fixed external screen, the device renders step-by-step procedural instructions, images, and animated content at the workstation while preserving operator attention and mobility throughout task execution on the shop floor.

The HL2 was selected for this role due to its technical maturity and its established adoption in AR-based HRC research. As reported by Costa, Petry, and Moreira [9], although discontinued, the HL1 and HL2 remain among the most commonly used HMDs in this domain to date, reflecting their widespread acceptance in both academia and industry. Moreover, the device supports hands-free operation and free movement across the shop floor, both of which are indispensable in industrial HRT scenarios where operators perform collaborative tasks alongside the robotic manipulators, in this thesis a UR5 .

The application deployed on the HL2 was developed in Unity using the Mixed Reality Toolkit (MRTK)[7], which enabled integration of spatial anchoring, user interface components, and input handling. Communication with the Robot Operating System (ROS) server is established via the *ROS-TCP Connector*, through which the device receives operation assignments and associated multimedia content as they are released by the scheduling framework. User interactions, including gesture inputs and task confirmations, are transmitted back to the system, enabling synchronous coordination between the operator and the robotic agent and effectively integrating the operator as an active element within the manufacturing loop. Spatial registration of virtual content onto the physical workstation is handled through the native QR code tracking algorithm, described in Section 4.4, which ensures stable anchoring as the operator moves within the environment.

To support this functionality, the HL2 integrates a comprehensive sensor suite, including visible-light and infrared cameras, a depth sensor, grayscale tracking cameras, and an IMU, as shown in Figure 4.4. Together, these sensors and the underlying hardware–software ecosystem enable real-time SLAM, spatial mapping, six-degrees-of-freedom tracking, and natural interaction modalities such as hand tracking, eye tracking, and voice commands, collectively providing the stable, low-latency registration of virtual content required in this collaborative industrial context [9], [102], [151]–[154]. Although the HL2 has been widely studied in AR-based HRC applications, its relevance in this work lies in its role as the operational interface enabling bidirectional communication between human operators and the manufacturing system. The tight coupling between instruction delivery and user feedback that positions it as a central component of the proposed architecture rather than a standalone visualization tool.

[7] `https://github.com/MixedRealityToolkit/`

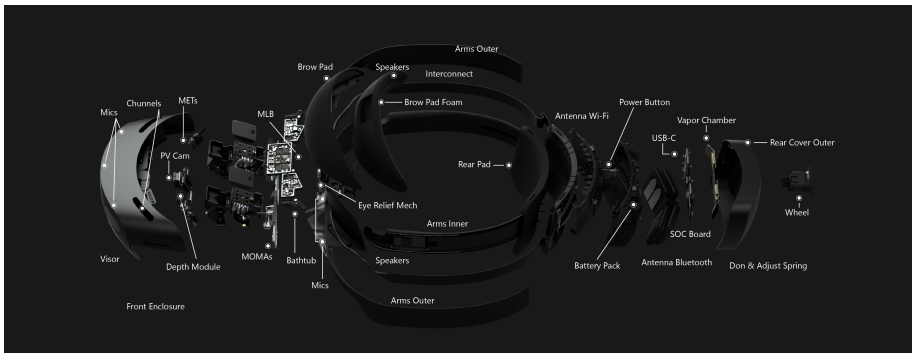


Figure 4.4: Microsoft HL 2 HMD used in this work, illustrating the wearable AR device that enables hands-free interaction and visualization of digital information within the operator's field of view. Adapted from Microsoft [154].

4.4 Tracking Algorithm

Referring back to Section 2.4, where virtual content and physical elements alignment strategies were discussed, for this thesis application, the marker-based approach was adopted due the aforementioned advantages. As the HL2 enables operators to move freely across the shop floor, holograms may gradually drift from their original anchor points. Marker-based systems allow fast and reliable re-calibration while generally requiring less computational power than markerless solutions [9]. However, committing to a specific tracking algorithm without an empirical basis would introduce uncertainty into the system, given that the choice of algorithm directly determines the spatial accuracy with which virtual instructions are registered onto the physical workspace. Imprecise registration would undermine the premise of spatially grounded AR support. To address this, a dedicated comparative study was conducted and published in Costa, Petry, Martins, *et al.* [17], evaluating four of the most widely used fiducial marker tracking algorithms available for the HL2, ARToolKit [16], ArUco [89], QR Codes[8] [155], and Vuforia[9].

4.4.1 Methodology of the Comparative Study

A key methodological concern in evaluating tracking algorithms is the elimination of human measurement error, which has affected many prior works in this area. The study addressed this by relying on the OptiTrack[10] high-accuracy motion capture system as ground truth, an approach that produces objective and reproducible measurements independent of observer judgment. The OptiTrack setup comprised twelve Prime X 22 cameras distributed across a capture volume of approximately $8.3 \times 5.5 \times 3.0\,\text{m}$, calibrated with Motive 3.0 and the OptiTrack calibration wand

[8] https://github.com/microsoft/MixedReality-QRCode-Sample
[9] https://developer.vuforia.com/home
[10] https://optitrack.com/

(CW-500), achieving a mean ray error of 0.592 mm[11] and a mean wand error of 0.155 mm[12]. Figure 4.5 shows the experiment setup.

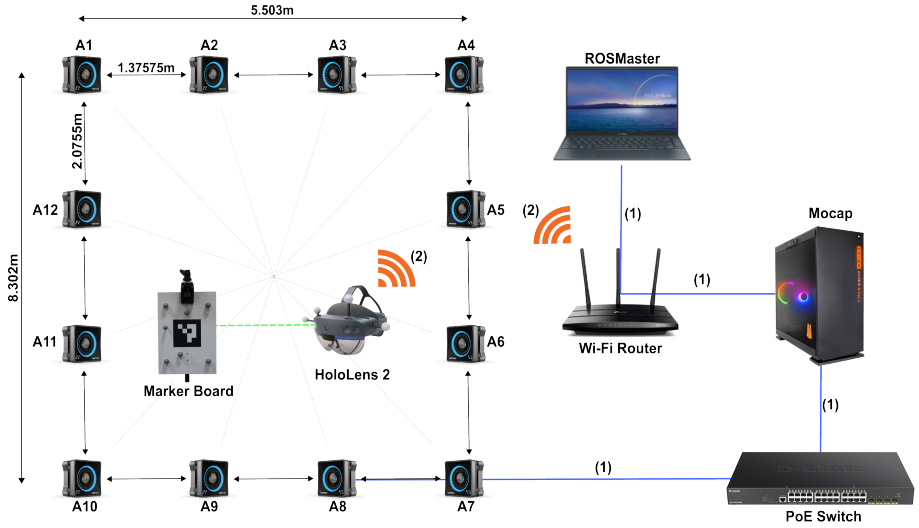


Figure 4.5: Fiducial Markers Experiment architecture and setup: (1) Ethernet cable connection; (2) Wireless connection.

Each fiducial marker was laser printed on a customized acrylic board manufactured to fit passive retroreflective spheres, allowing OptiTrack to track the board's pose with sub-millimeter accuracy. Similar spheres were attached to the HL2, which was kept static while the marker board was positioned at 25 sampling points (Figure 4.7), defined by five distances (300, 600, 900, 1200, and 1500 mm) and five angular orientations (-30º, -15º, 0º, 15º, and 30º) relative to the HL2 Photo/Video camera. At each sampling point, 500 consecutive pose observations were recorded to estimate both the mean tracking error and its standard deviation, providing a robust measure of accuracy and precision respectively.

The tracking error was computed as the transformation between the pose estimated by each algorithm and the ground truth pose derived from OptiTrack, decomposed into translational error (in mm) and rotational error (in degrees) along each axis. Applications using the HL2 Photo/Video camera were set to a resolution of 1920 x 1080 pixels, while the QR Code algorithm, which uses the HL2 stereo cameras, operated at its native resolution of 640 x 480 pixels. Figure 4.6 shows the transformations between the fiducial marker and the HL2, while Figure 4.7 shows the sampling poses.

4.4.2 Results and Discussion

Figures 4.8, 4.9, 4.10, and 4.11 depict the translational and rotational errors in three-dimensional space for each algorithm, and Tables 4.1, 4.2, 4.3, 4.4 report the per-axis mean errors and standard deviations at each sampling point. A finding common to all four algorithms is that translational

[11]Represents the preciseness of the calculated 3D points during calibration.
[12]Reports the error of the detected wand length compared to the expected wand length throughout the calibration.

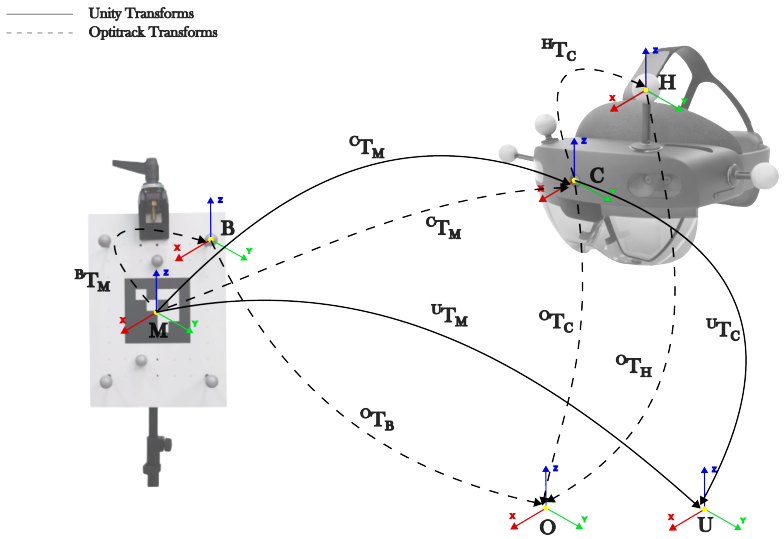


Figure 4.6: Definition of the spatial transformation between the HL2 and the fiducial marker: Continuous lines indicate transformations provided by our Unity app; Dotted lines indicate transformations provided by Optitrack. The board (B) and marker (M) referential frames are coincident. However, for better visualization and interpretation, they are represented separately.

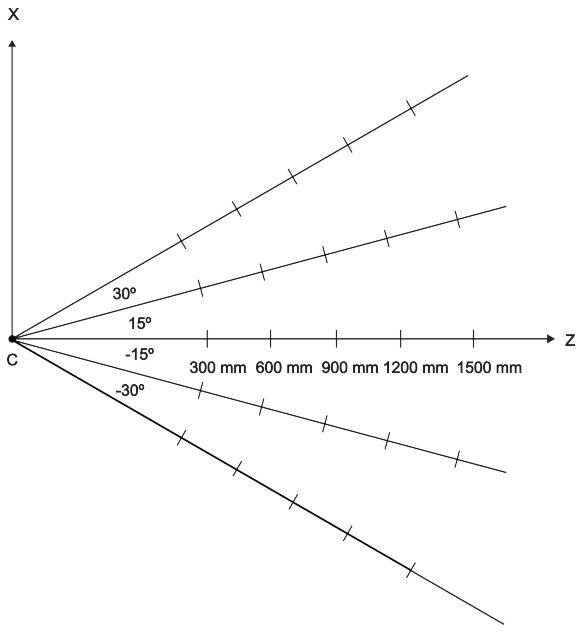


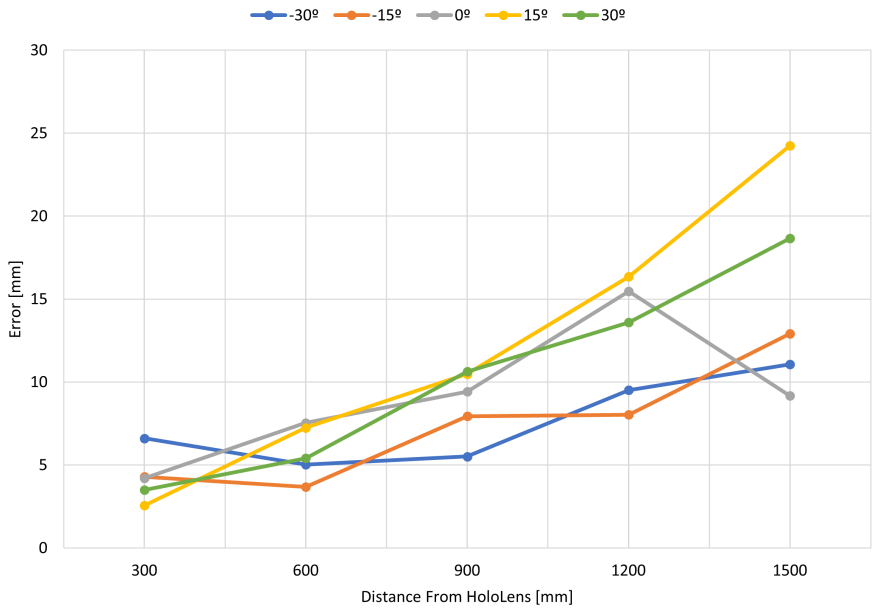
Figure 4.7: Experiment's workspace: The HL2 Photo/Video Camera (*C*) was positioned in the origin of the reference frame, and the board in each of the 25 sampling points. Sampling points range from 300 to 1500 mm and from -30° to 30°. The board is always perpendicular to the line connecting the sampling point and *C*.

error increases with distance from the HL2, regardless of marker orientation. Notably, significant spikes are visible at [300 mm; +30°] in the ARToolKit results, at [300 mm; ±30°] in the ArUco results, and at [900 mm; 0°] in the QR Code results. In those cases, the respective algorithms failed to detect the fiducial marker pattern reliably at oblique angles or at the extended distance; because the estimation error was consistent across all 500 observations at those points, recollecting the samples would not resolve the issue.

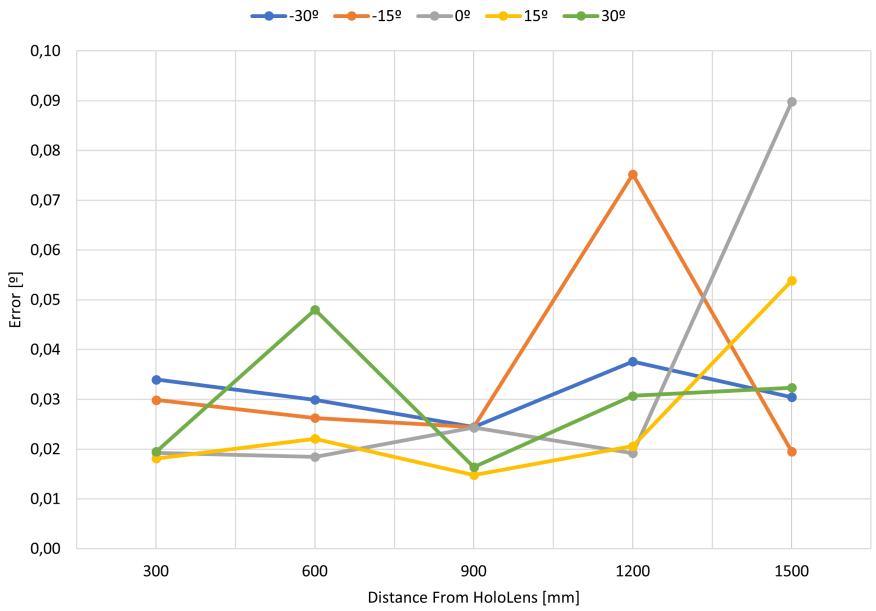
Regarding Vuforia (Figure 4.8 and Table 4.1), negative board angular directions produced lower translational errors than positive ones. On the X-axis (width), errors shifted toward increasingly positive values with distance across all angular directions; on the Y-axis (height), positive directions trended upward with distance while negative directions trended downward; and on the Z-axis (depth), errors increased in magnitude in the negative direction as the board moved further from the HL2. Rotational errors remained within a narrow range across all angular directions, with no clearly dominant axis-specific pattern. The exception was the Z-axis (roll), where the lowest and highest negative values were observed at -15 and -30, respectively.

Table 4.1: Mean translation and rotation errors of the Vuforia marker tracker along the X, Y, and Z axes, reported as mean ± standard deviation, for all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$)

Angle	**Distance**	**Translation Error (mm)**			**Rotation Error(°)**		
		X	**Y**	**Z**	**θX**	**θY**	**θZ**
-30°	300 mm	-6.34 ± 0.13	-1.75 ± 0.08	0.47 ± 0.06	0.023 ± 0.000	0.011 ± 0.000	-0.021 ± 0.000
	600 mm	-4.66 ± 0.08	-1.82 ± 0.14	0.45 ± 0.24	0.018 ± 0.000	0.018 ± 0.001	-0.015 ± 0.000
	900 mm	-3.84 ± 0.10	-2.60 ± 0.16	-2.96 ± 0.10	0.001 ± 0.001	0.005 ± 0.004	-0.023 ± 0.000
	1200 mm	-3.44 ± 0.12	-4.00 ± 0.28	-7.90 ± 0.18	-0.024 ± 0.001	-0.021 ± 0.001	-0.019 ± 0.000
	1500 mm	-1.57 ± 0.11	-3.99 ± 0.20	-10.18 ± 0.25	-0.027 ± 0.002	-0.004 ± 0.002	-0.011 ± 0.000
-15°	300 mm	-4.03 ± 0.12	-0.79 ± 0.19	1.16 ± 0.14	0.002 ± 0.000	-0.029 ± 0.001	-0.006 ± 0.000
	600 mm	-2.35 ± 0.14	-0.89 ± 0.17	-2.67 ± 0.12	-0.010 ± 0.000	0.023 ± 0.000	-0.006 ± 0.000
	900 mm	-2.16 ± 0.04	-1.77 ± 0.10	-7.42 ± 0.02	-0.023 ± 0.000	-0.004 ± 0.000	-0.004 ± 0.000
	1200 mm	-0.02 ± 0.17	-1.32 ± 0.17	-7.91 ± 0.15	0.010 ± 0.000	-0.074 ± 0.000	-0.003 ± 0.000
	1500 mm	-0.96 ± 0.12	-2.31 ± 0.18	-12.67 ± 0.20	-0.013 ± 0.001	0.000 ± 0.001	-0.014 ± 0.000
0°	300 mm	-3.07 ± 0.09	-1.26 ± 0.14	-2.57 ± 0.10	-0.001 ± 0.000	0.013 ± 0.003	-0.013 ± 0.000
	600 mm	-2.12 ± 0.03	-0.92 ± 0.07	-7.17 ± 0.02	-0.003 ± 0.000	0.005 ± 0.000	-0.017 ± 0.000
	900 mm	0.08 ± 0.14	0.07 ± 0.22	-9.41 ± 0.22	-0.009 ± 0.000	0.018 ± 0.001	-0.012 ± 0.000
	1200 mm	1.40 ± 0.22	0.10 ± 0.23	-15.39 ± 0.42	-0.007 ± 0.001	0.009 ± 0.001	-0.014 ± 0.000
	1500 mm	4.00 ± 0.16	4.47 ± 0.32	-6.85 ± 1.41	0.085 ± 0.028	0.013 ± 0.009	-0.021 ± 0.001
15°	300 mm	-1.46 ± 0.10	-1.15 ± 0.19	-1.74 ± 0.16	0.000 ± 0.001	-0.001 ± 0.012	-0.013 ± 0.000
	600 mm	1.53 ± 0.10	-0.99 ± 0.16	-7.00 ± 0.34	-0.008 ± 0.000	0.016 ± 0.001	-0.011 ± 0.000
	900 mm	4.78 ± 0.04	-0.20 ± 0.12	-9.32 ± 0.02	-0.004 ± 0.000	-0.004 ± 0.000	-0.013 ± 0.000
	1200 mm	7.41 ± 0.20	-0.21 ± 0.28	-14.56 ± 0.25	-0.007 ± 0.001	0.011 ± 0.001	-0.014 ± 0.000
	1500 mm	10.32 ± 0.24	-0.31 ± 1.15	-21.83 ± 3.61	-0.029 ± 0.041	-0.025 ± 0.005	-0.019 ± 0.000
30°	300 mm	-1.91 ± 0.01	-0.52 ± 0.06	-2.86 ± 0.02	0.010 ± 0.000	-0.013 ± 0.000	-0.009 ± 0.000
	600 mm	1.88 ± 0.11	-0.23 ± 0.20	-5.05 ± 0.29	0.006 ± 0.002	-0.046 ± 0.001	-0.008 ± 0.000
	900 mm	5.86 ± 0.04	0.47 ± 0.11	-8.86 ± 0.02	0.005 ± 0.000	-0.006 ± 0.000	-0.014 ± 0.000
	1200 mm	9.25 ± 0.12	1.46 ± 0.20	-9.84 ± 0.03	-0.010 ± 0.000	0.025 ± 0.000	-0.014 ± 0.000
	1500 mm	14.49 ± 0.10	2.11 ± 0.25	-11.55 ± 0.51	0.006 ± 0.003	-0.027 ± 0.000	-0.014 ± 0.000



(a) Vuforia Translational error (mm) across all tested distances (300–1500 mm) and angular directions (-30° to 30°)



(b) Vuforia Orientation error ($^\circ$) across all tested distances (300–1500 mm) and angular directions (-30° to 30°)

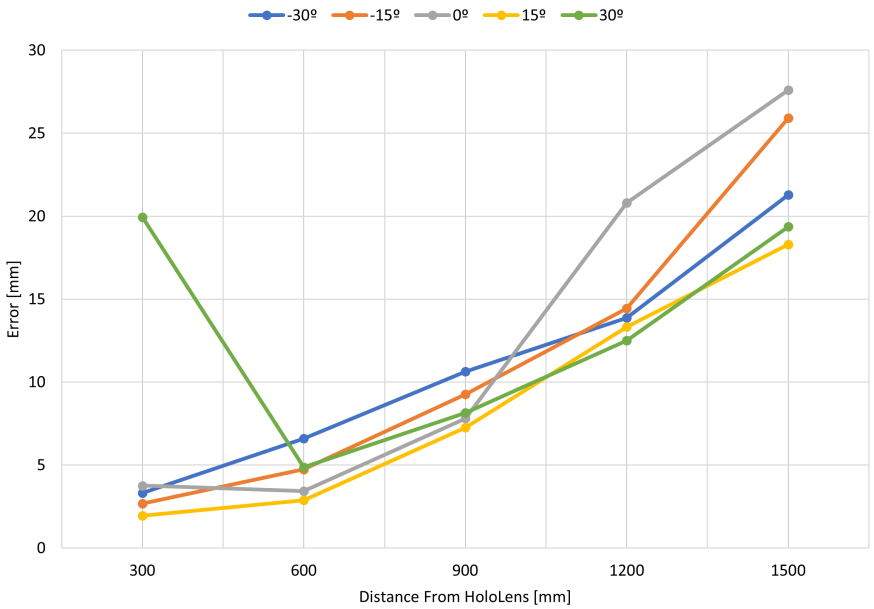
Figure 4.8: Vuforia marker tracking errors across all tested distances (300–1500 mm) and angular directions (-30° to 30°): (a) translational error (mm); (b) orientation error ($^\circ$)

Regarding ARToolKit (Figure 4.9 and Table 4.2), positive angular directions yielded lower translational errors than negative ones, with an upward trend across all three axes as the board moved further from the HL2, and the closer the board was, the smaller its contribution to the overall error. Rotational errors were generally stable, except at 0° and 1500 mm, where a pronounced outlier is visible. On the X-axis, all directions showed a downward trend toward negative values with increasing distance; on the Y-axis, a corresponding upward trend was observed; on the Z-axis, errors remained within a narrow band across all conditions.

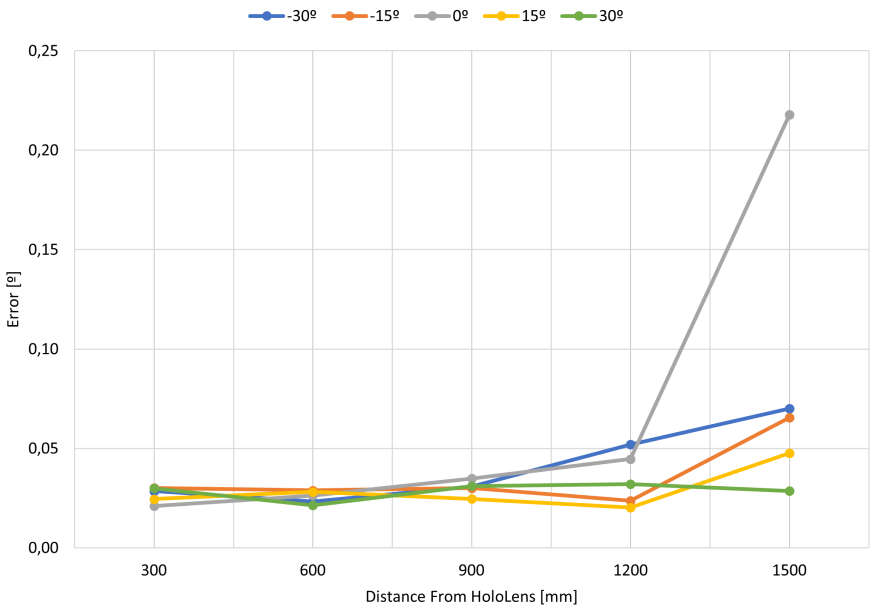
Table 4.2: Mean translation and rotation errors of the ARToolKit marker tracker along the X, Y, and Z axes, reported as mean ± standard deviation, for all tested distances (300–1500 mm) and angular directions ($-30°$ to 30°)

Angle	**Distance**	**Translation Error (mm)**			**Rotation Error(°)**		
		X	**Y**	**Z**	**θX**	**θY**	**θZ**
-30°	300 mm	-2.52 ± 0.05	0.01 ± 0.07	2.14 ± 0.05	0.013 ± 0.000	-0.019 ± 0.000	-0.015 ± 0.000
	600 mm	1.73 ± 0.07	1.70 ± 0.09	6.13 ± 0.04	-0.011 ± 0.000	0.010 ± 0.000	-0.016 ± 0.000
	900 mm	4.91 ± 0.05	2.83 ± 0.10	8.97 ± 0.04	-0.021 ± 0.000	0.011 ± 0.000	-0.018 ± 0.000
	1200 mm	7.66 ± 0.07	4.05 ± 0.13	10.81 ± 0.06	-0.047 ± 0.000	0.016 ± 0.000	-0.013 ± 0.000
	1500 mm	10.22 ± 0.07	6.05 ± 0.15	17.66 ± 0.05	-0.060 ± 0.000	0.031 ± 0.000	-0.018 ± 0.000
-15°	300 mm	-2.62 ± 0.21	-0.46 ± 0.12	-0.23 ± 0.12	0.001 ± 0.002	-0.017 ± 0.019	-0.014 ± 0.001
	600 mm	1.16 ± 0.05	1.85 ± 0.09	4.19 ± 0.06	-0.014 ± 0.000	0.017 ± 0.000	-0.017 ± 0.000
	900 mm	3.84 ± 0.05	3.15 ± 0.11	7.81 ± 0.08	-0.017 ± 0.000	0.015 ± 0.000	-0.018 ± 0.000
	1200 mm	6.21 ± 0.12	4.78 ± 0.19	12.11 ± 0.26	-0.012 ± 0.003	0.009 ± 0.003	-0.017 ± 0.001
	1500 mm	8.21 ± 1.09	7.49 ± 1.44	23.36 ± 6.31	-0.019 ± 0.002	0.057 ± 0.027	-0.018 ± 0.000
0°	300 mm	-2.03 ± 0.05	-0.50 ± 0.07	-3.11 ± 0.08	0.003 ± 0.000	-0.010 ± 0.000	-0.017 ± 0.000
	600 mm	1.00 ± 0.12	0.44 ± 0.19	-3.25 ± 0.10	-0.014 ± 0.000	0.010 ± 0.001	-0.019 ± 0.000
	900 mm	3.84 ± 0.05	3.92 ± 0.13	5.54 ± 0.06	-0.017 ± 0.000	0.026 ± 0.000	-0.014 ± 0.000
	1200 mm	6.80 ± 0.06	7.97 ± 0.19	17.95 ± 0.10	-0.040 ± 0.000	0.001 ± 0.000	-0.018 ± 0.000
	1500 mm	6.49 ± 0.06	4.29 ± 0.19	26.48 ± 0.30	0.179 ± 0.001	-0.122 ± 0.001	-0.003 ± 0.000
15°	300 mm	-1.12 ± 0.08	-0.26 ± 0.11	-1.55 ± 0.18	0.011 ± 0.001	-0.011 ± 0.002	-0.017 ± 0.001
	600 mm	2.05 ± 0.04	1.32 ± 0.09	-1.51 ± 0.05	-0.015 ± 0.000	0.019 ± 0.000	-0.013 ± 0.000
	900 mm	4.01 ± 0.06	4.65 ± 0.15	3.86 ± 0.06	-0.019 ± 0.000	0.001 ± 0.000	-0.014 ± 0.000
	1200 mm	5.29 ± 0.06	9.31 ± 0.19	7.92 ± 0.06	-0.016 ± 0.000	-0.003 ± 0.000	-0.011 ± 0.000
	1500 mm	6.88 ± 0.07	12.11 ± 0.24	11.84 ± 0.06	-0.017 ± 0.000	0.042 ± 0.001	-0.012 ± 0.000
30°	300 mm	-7.59 ± 11.33	3.88 ± 6.38	-3.52 ± 12.02	-0.011 ± 0.004	0.022 ± 0.016	0.002 ± 0.012
	600 mm	0.18 ± 0.10	4.22 ± 0.16	2.41 ± 0.18	-0.001 ± 0.000	0.017 ± 0.000	-0.011 ± 0.000
	900 mm	2.93 ± 0.04	6.87 ± 0.12	3.26 ± 0.05	-0.018 ± 0.000	0.022 ± 0.000	-0.010 ± 0.000
	1200 mm	4.58 ± 0.06	9.57 ± 0.17	6.60 ± 0.06	-0.027 ± 0.000	0.010 ± 0.000	-0.012 ± 0.000
	1500 mm	7.03 ± 0.10	12.92 ± 0.23	12.58 ± 0.08	-0.017 ± 0.000	0.020 ± 0.000	-0.009 ± 0.000

Regarding ArUco (Figure 4.10 and Table 4.3), translational error on the X-axis trended toward positive values with increasing distance for all angular directions. On the Y-axis, the influence of distance was most pronounced at -15 and least at -30. On the Z-axis, downward tendencies were observed for the -30 and 15 directions. Rotational errors were generally stable, apart from an outlier at 0° and 1200 mm, with a slight upward drift toward positive values as the marker moved further. On the Z-axis, a minor measurement at 30° and 300 mm fell outside the general band, but its small magnitude had negligible influence on the overall trend.

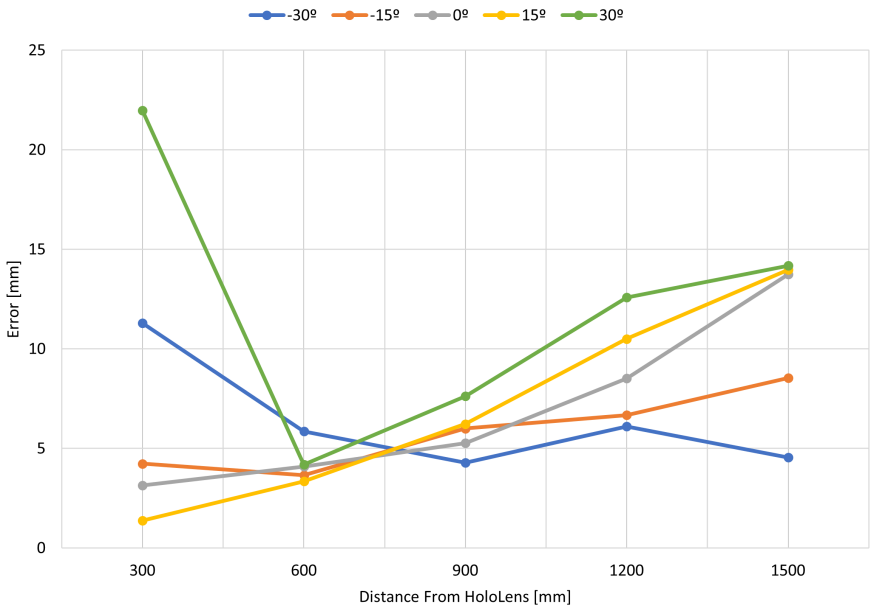


(a) ARToolKit Translational error (mm) across all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$)

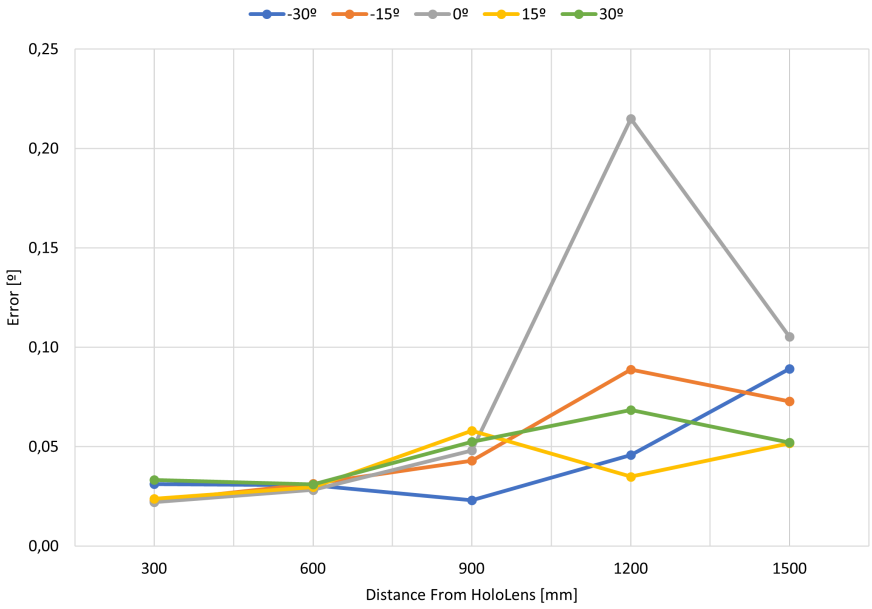


(b) ARToolKit Orientation error (°) across all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$)

Figure 4.9: ARToolKit marker tracking errors across all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$): (a) translational error (mm); (b) orientation error (°)



(a) ArUco Translational error (mm) across all tested distances (300–1500 mm) and angular directions (−30° to 30°)



(b) ArUco Orientation error (°) across all tested distances (300–1500 mm) and angular directions (−30° to 30°)

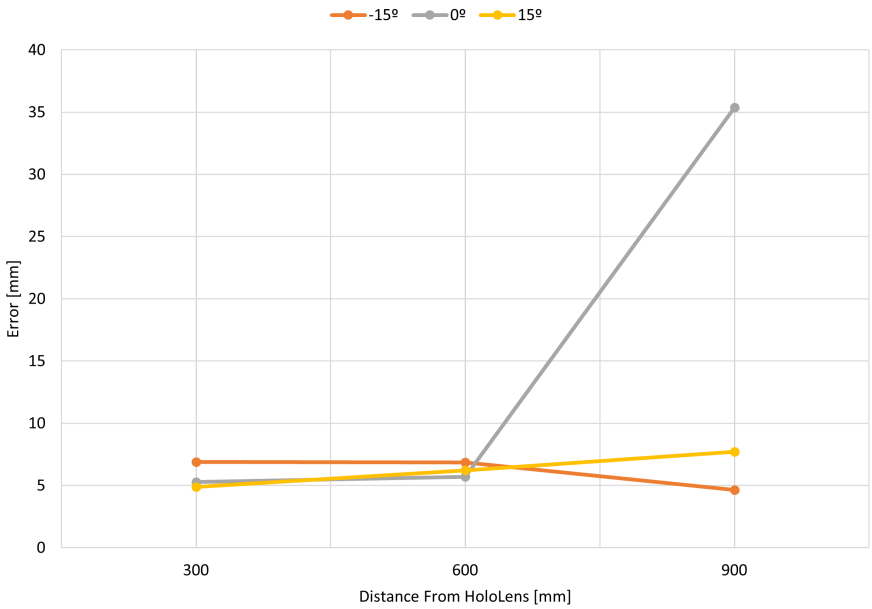
Figure 4.10: ArUco marker tracking errors across all tested distances (300–1500 mm) and angular directions (−30° to 30°): (a) translational error (mm); (b) orientation error (°)

Table 4.3: Mean translation and rotation errors of the ArUco marker tracker along the X, Y, and Z axes, reported as mean ± standard deviation, for all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$)

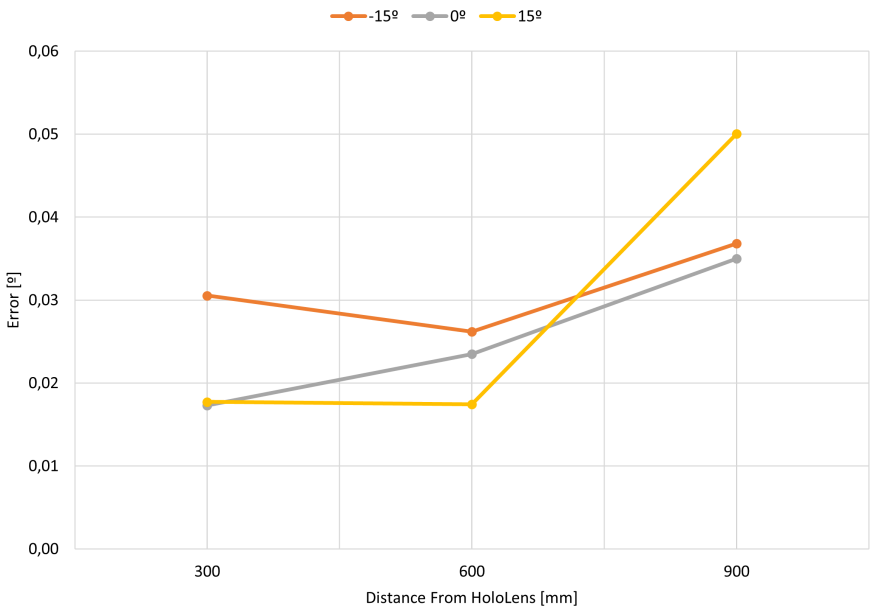
Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	**Y**	**Z**	**θX**	**θY**	**θZ**
-30°	300 mm	5.50 ± 0.03	-0.65 ± 0.03	9.82 ± 0.03	-0.000 ± 0.000	-0.001 ± 0.000	-0.031 ± 0.000
	600 mm	-1.44 ± 0.15	-1.52 ± 0.08	5.45 ± 0.46	0.010 ± 0.007	-0.023 ± 0.011	-0.014 ± 0.001
	900 mm	-0.32 ± 0.08	-1.71 ± 0.15	3.86 ± 1.09	-0.009 ± 0.012	-0.002 ± 0.006	-0.016 ± 0.001
	1200 mm	0.64 ± 0.15	-1.50 ± 0.42	5.83 ± 1.69	-0.007 ± 0.055	0.010 ± 0.005	-0.019 ± 0.003
	1500 mm	1.99 ± 0.28	-2.05 ± 0.52	0.77 ± 3.58	-0.053 ± 0.079	0.001 ± 0.025	-0.022 ± 0.005
-15°	300 mm	-2.11 ± 0.02	-1.76 ± 0.05	3.20 ± 0.04	0.001 ± 0.001	-0.016 ± 0.000	-0.015 ± 0.000
	600 mm	-1.10 ± 0.03	-3.26 ± 0.06	-1.22 ± 0.03	-0.022 ± 0.001	0.014 ± 0.002	-0.016 ± 0.000
	900 mm	0.86 ± 0.05	-5.06 ± 0.19	-2.90 ± 1.22	-0.031 ± 0.015	0.001 ± 0.003	-0.026 ± 0.000
	1200 mm	3.36 ± 0.10	-5.41 ± 0.20	-0.36 ± 1.95	0.043 ± 0.089	0.011 ± 0.006	-0.018 ± 0.002
	1500 mm	6.18 ± 0.18	-4.80 ± 0.36	2.79 ± 1.90	-0.048 ± 0.019	0.050 ± 0.016	-0.008 ± 0.001
0°	300 mm	-2.05 ± 0.03	-2.36 ± 0.04	-0.07 ± 0.10	0.003 ± 0.003	0.002 ± 0.001	-0.021 ± 0.000
	600 mm	0.18 ± 0.07	-3.13 ± 0.09	-2.59 ± 0.26	-0.012 ± 0.002	-0.000 ± 0.006	-0.024 ± 0.001
	900 mm	5.18 ± 0.03	-0.49 ± 0.20	-0.72 ± 0.14	-0.015 ± 0.015	0.041 ± 0.002	-0.012 ± 0.001
	1200 mm	7.52 ± 0.18	-0.19 ± 0.67	3.38 ± 2.25	0.171 ± 0.113	-0.075 ± 0.077	-0.004 ± 0.004
	1500 mm	10.98 ± 0.10	-1.83 ± 0.99	-7.04 ± 4.40	0.029 ± 0.099	0.037 ± 0.017	-0.022 ± 0.001
15°	300 mm	-0.61 ± 0.14	-1.14 ± 0.08	0.32 ± 0.25	0.013 ± 0.003	-0.007 ± 0.002	-0.017 ± 0.001
	600 mm	2.70 ± 0.05	-1.08 ± 0.09	-1.61 ± 0.39	0.008 ± 0.007	-0.012 ± 0.019	-0.015 ± 0.000
	900 mm	6.03 ± 0.14	-0.46 ± 0.20	-0.59 ± 1.32	0.040 ± 0.030	-0.019 ± 0.022	-0.017 ± 0.002
	1200 mm	10.03 ± 0.17	-0.01 ± 0.38	-2.51 ± 1.81	0.003 ± 0.011	-0.023 ± 0.009	-0.021 ± 0.001
	1500 mm	12.39 ± 0.15	0.91 ± 0.36	-6.00 ± 2.19	-0.033 ± 0.023	0.017 ± 0.024	-0.019 ± 0.000
30°	300 mm	-21.79 ± 0.04	-2.40 ± 0.06	1.34 ± 0.08	-0.024 ± 0.000	0.018 ± 0.000	0.012 ± 0.000
	600 mm	3.90 ± 0.05	-0.64 ± 0.08	1.25 ± 0.49	0.013 ± 0.008	-0.022 ± 0.002	-0.013 ± 0.000
	900 mm	7.55 ± 0.12	-0.13 ± 0.19	-0.34 ± 0.90	-0.029 ± 0.012	0.032 ± 0.018	-0.019 ± 0.001
	1200 mm	12.04 ± 0.15	1.26 ± 0.30	3.18 ± 1.25	-0.056 ± 0.021	0.026 ± 0.015	-0.018 ± 0.002
	1500 mm	12.83 ± 0.23	-0.66 ± 0.31	4.70 ± 3.74	0.012 ± 0.025	-0.027 ± 0.028	-0.019 ± 0.001

Regarding the QR Code algorithm (Figure 4.11 and Table 4.4), translational errors remained within a narrow band across angular directions, with the exception of the 0° and 900 mm outlier. On the X-axis, all directions showed an upward trend, accelerating notably at 900 mm; on the Y-axis, errors increased toward positive values; and on the Z-axis, a slight downward tendency was observed despite overall stability. Rotational errors were stable up to 600 mm and increased at further distances. On the Y-axis, a downward tendency toward negative values was strongest at 15° and 900 mm. The algorithm could not reliably detect the marker at far distances and oblique angles, attributable to the HL2 stereo cameras' limited field of view and lower resolution compared to the Photo/Video camera.

Figures 4.12 and 4.13 map, across the experiment's workspace, the best-performing algorithm at each sampling position in terms of accuracy and precision for translation and rotation, respectively. From a distance-wise perspective, ArUco was the most translationally accurate at all measured distances. ARToolKit had good translational accuracy at short range (300–600 mm) but was the worst at further distances. Vuforia showed low translational accuracy at mid-range dis-



(a) QR Code Translational error (mm) across all tested distances (300–1500 mm) and angular directions (-30° to 30°)



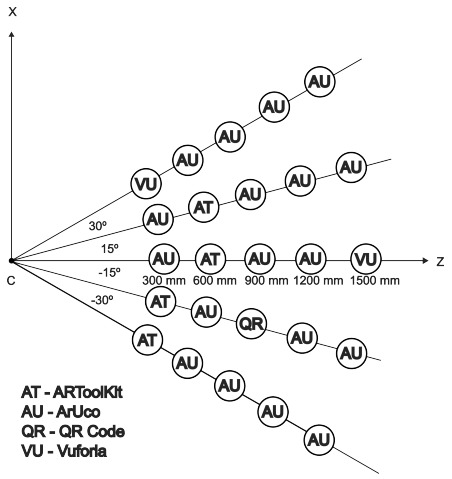
(b) QR Code Orientation error ($^\circ$) across all tested distances (300–1500 mm) and angular directions (-30° to 30°)

Figure 4.11: QR Code marker tracking errors across all tested distances (300–1500 mm) and angular directions (-30° to 30°): (a) translational error (mm); (b) orientation error ($^\circ$)

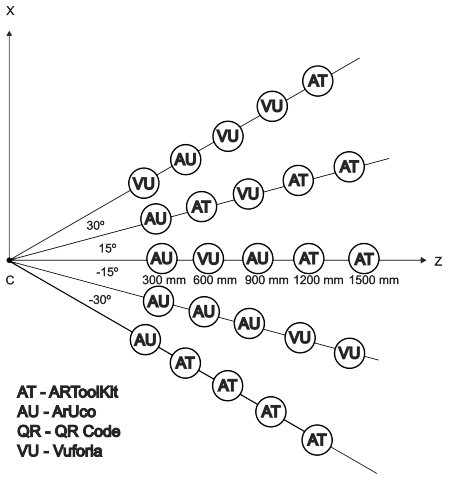
Table 4.4: Mean translation and rotation errors of the QR Code marker tracker along the X, Y, and Z axes, reported as mean ± standard deviation, for all tested distances (300–1500 mm) and angular directions ($-30°$ to $30°$)

Angle	Distance	Translation Error (mm)			Rotation Error(°)		
		X	**Y**	**Z**	**θX**	**θY**	**θZ**
-15°	300 mm	-1.33 ± 0.14	1.56 ± 0.15	6.56 ± 0.17	0.014 ± 0.000	-0.026 ± 0.001	-0.004 ± 0.000
	600 mm	-0.32 ± 0.17	2.55 ± 0.17	6.34 ± 0.15	0.001 ± 0.000	-0.023 ± 0.008	-0.006 ± 0.000
	900 mm	1.72 ± 0.20	2.45 ± 0.37	0.89 ± 3.86	0.014 ± 0.005	-0.032 ± 0.002	-0.006 ± 0.000
0°	300 mm	-3.07 ± 0.18	0.83 ± 0.17	4.20 ± 0.22	0.008 ± 0.002	-0.002 ± 0.004	-0.014 ± 0.000
	600 mm	-0.87 ± 0.11	2.45 ± 0.14	5.05 ± 0.25	0.016 ± 0.002	-0.010 ± 0.004	-0.012 ± 0.000
	900 mm	9.74 ± 0.12	2.97 ± 0.25	-33.86 ± 0.28	0.025 ± 0.001	-0.022 ± 0.003	-0.007 ± 0.001
15°	300 mm	-2.97 ± 0.12	1.20 ± 0.16	3.67 ± 0.16	0.012 ± 0.001	0.004 ± 0.004	-0.010 ± 0.000
	600 mm	-0.80 ± 0.12	3.35 ± 0.21	5.16 ± 0.25	0.013 ± 0.003	-0.001 ± 0.004	-0.008 ± 0.000
	900 mm	6.73 ± 0.09	2.21 ± 0.28	3.00 ± 0.17	0.017 ± 0.000	-0.045 ± 0.000	-0.009 ± 0.000

tances (600–1200 mm), with peak inaccuracy at 600 mm, while the QR Code presented the worst translational accuracy at 300 mm. Despite leading in translational accuracy, ArUco was the most precise only at 300 mm and the worst in precision from 900 to 1500 mm. Vuforia was the most precise at 900 mm but the worst at 300 and 600 mm. Regarding rotation, ArUco was the most inaccurate at all distances, and Vuforia was the most accurate from 900 to 1500,mm. In terms of precision, ARToolKit was the most precise at all distances, with Vuforia also showing consistently high precision. The QR Code was the most imprecise at 300,mm, despite being the most accurate at that distance, and ArUco was the most imprecise from 600 to 1500,mm.



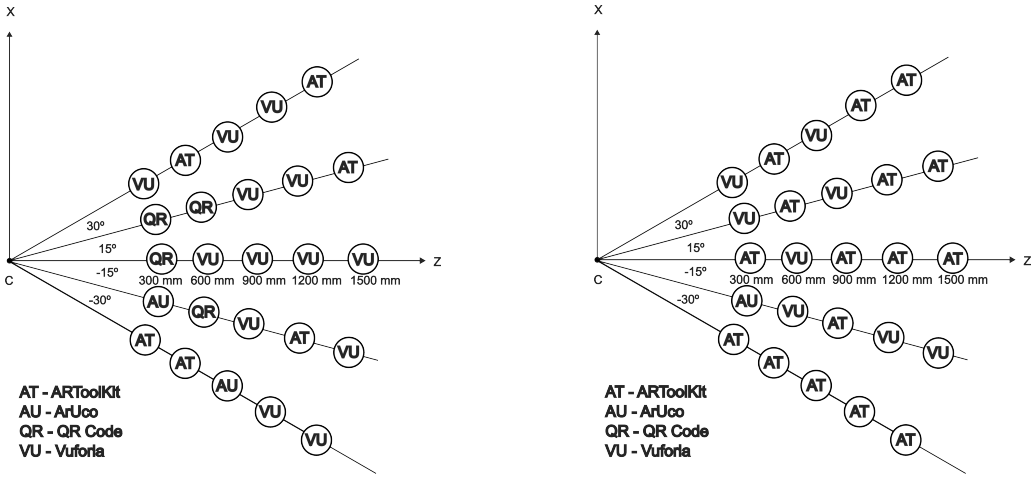
(a) Most accurate algorithm for translational estimation at each sampled position across the tested workspace.



(b) Most precise algorithm for translational estimation at each sampled position across the tested workspace.

Figure 4.12: Spatial representation of the best-performing algorithms for translational estimation at each sampled position (distance and angular direction): (a) accuracy; (b) precision.

From an angular direction-wise perspective, ArUco maintained the best translational accuracy



(a) Most accurate algorithm for rotational estimation at each sampled position across the tested workspace.

(b) Most precise algorithm for rotational estimation at each sampled position across the tested workspace.

Figure 4.13: Spatial representation of the best-performing algorithms for rotational estimation at each sampled position (distance and angular direction): (a) accuracy; (b) precision.

across all tested orientations, while ARToolKit was the most inaccurate from -30 to 0 and Vuforia was the most inaccurate at -15 and 30. In terms of translational precision by angle, ArUco was the most precise at -15 and 0, but the most imprecise at -30 and 30. ARToolKit was the most precise at 15, and Vuforia was the most precise at 30. Regarding rotational accuracy by angular direction, ArUco was the most inaccurate in all directions, and Vuforia was the most accurate overall. For rotational precision, ARToolKit was the most precise across the majority of angular directions, while ArUco was consistently the least precise.

In summary, ArUco is the most accurate algorithm for translational estimation, while ARToolKit and Vuforia were the most inaccurate in that respect. Conversely, ARToolKit and Vuforia are the most precise. For rotational estimation, ArUco was the most inaccurate and imprecise overall, whereas ARToolKit was the most precise and Vuforia and QR Code were the most accurate. Table 4.5 summarizes the strengths and weaknesses of each algorithm.

These results are further contextualized by comparison with prior work using the same algorithms on the HL2. Regarding ArUco, Doughty and Ghugre [89] reported a translational error of 1.02 mm on the XY plane at approximately 635 mm using a point-and-trace 2D procedure, while this study estimated 3.14 mm at 600 mm in Euclidean space, reflecting the additional error introduced by the Z-axis depth component. Gsaxner, Li, Pepe, *et al.* [88] reported mean RMS errors of 6.09 mm and 6.73° for distances from 200 to 800 mm; the present study obtained average RMS errors of 6.90 mm and 0.035° between 300 and 900 mm, underperforming in translation but substantially improving on rotational accuracy. Comparison with Thabit, Niessen, Wolvius, *et al.* [99], using an 80 mm marker, shows broadly similar translational performance across axes, with this study performing better on rotation at all axes.

Regarding ARToolKit, the average translational error obtained here at 600 mm (4.507 mm in

Table 4.5: Qualitative comparison of the four evaluated marker tracking algorithms in terms of estimation accuracy, precision, implementation requirements, and HL2 hardware constraints.

Tracking algorithm	Strengths	Weaknesses
ARToolKit	Precise translation and rotation estimation Open Source Large marker library Does not block HL2 Photo/Video camera use	Inaccurate translation estimation Requires deep programming knowledge for implementations
ArUco	Accurate translation estimation Open Source Large marker library Does not block HL2 Photo/Video camera use	Inaccurate rotation estimation Requires deep programming knowledge for implementations
Vuforia	Good rotation estimation 2D and 3D tracking options Not restricted to a library Frequently updated SDK and active support Does not requires deep programming knowledge for implementations	Inaccurate translation estimation Block HL2 Photo/Video camera use Proprietary software
QR Code	Native support in HL2 Accurate rotation estimation Does not require deep programming knowledge for implementations	Limited resolution and working distance Does not support QR codes versions above 10 and with logos

3D, 2.48 mm on the XY plane) is comparable to the 4.27 mm 2D overlay error reported by Qian, Deguet, and Kazanzides [156], with this study performing better when restricted to the same plane. In comparison with the uncalibrated scenarios reported by Azimi, Qian, Navab, *et al.* [16] and Brand, Wulff, Hamdani, *et al.* [157] using head-anchored mono-camera approaches, this study's results were more accurate; however, against their calibrated scenarios, translational estimation underperformed. Eom, Kim, Rahimpour, *et al.* [158] evaluated the hologram translation error by measuring the image registration error with a caliper. This paper's approach performed better on the X and Y-axis translation estimation and underperformed on the Z-axis, at 600 mm.

Regarding Vuforia, this study used marker-based tracking while Eom, Kim, Rahimpour, *et al.* [158] and Frantz, Jansen, Duerinck, *et al.* [159] used Vuforia's 3D model tracking with different evaluation methodologies. Against Eom, Kim, Rahimpour, *et al.* [158], performance was lower on the X and Z axes but marginally better on the Y-axis. Against Frantz, Jansen, Duerinck, *et al.* [159], averaging errors over a comparable distance range (300–900 mm vs. their 400–800 mm), this study obtained 6.27 mm versus their 1.41 mm, which is partially attributable to the different angular conventions used in each experiment.

Regarding QR Code, the Microsoft stereo-camera implementation used here achieved an average error of 5.49 mm from 300 to 600 mm, outperforming the single-marker results reported by Kyaw, Xu, Jahn, *et al.* [160] using the HL2 front camera, though underperforming at 900 mm. With their linearization algorithm, Kyaw, Xu, Jahn, *et al.* [160] achieved errors as low as 0.97 mm, highlighting the potential gains from post-processing approaches.

4.4.3 Algorithm Selection and Implementation

The results provided a clear and principled basis for the tracking algorithm selection. The workstation at WS 10 operates within a close-range workspace where the operator interacts at distances well within the QR Code algorithm's effective detection range. In this context, the rotational stability of the QR Code algorithm is particularly valuable, as precise hologram orientation is critical

for the correct spatial alignment of virtual content with the physical robot and workspace elements, since misalignment would directly degrade the quality of the operator support the system is designed to provide. Furthermore, its native integration in the HL2 ecosystem eliminates additional software dependencies, reduces implementation complexity, and avoids the battery overhead associated with running a third-party tracking Software Development Kit (SDK), an important consideration for a system intended for prolonged use on the shop floor. The fact that it does not block the Photo/Video camera also preserves that resource for potential future extensions of the system.

The fiducial marker used in the implementation is a square QR code with a side length of 10 cm, placed at a known position within the workspace, with a predefined offset relative to the robot, providing a stable physical reference for spatial registration. As detailed in Section 6.4, an initial calibration step is performed as part of the tutorial designed to familiarize users with the AR interface. Given that the workstation remains static, continuous tracking of the QR code is not required to maintain alignment, a single calibration at startup is sufficient to register the virtual coordinate frame to the physical environment with the rotational precision demonstrated in Costa, Petry, Martins, *et al.* [17]. However, recalibration can be performed at any time, if necessary, by re-aligning the HL2 with the QR code. The marker position relative to the setup is shown in Figure 4.14



Figure 4.14: QR code used for pose estimation and spatial referencing within the collaborative workstation, providing a fixed reference for aligning virtual content with the physical environment.

4.5 Communication Framework

The communication framework enables the end-to-end pipeline described in Section 4.1: it carries the decisions of the coordination framework to the physical system and to the operator, and returns the state of the system and the operator's confirmations to the coordination framework. The architecture is presented here at the logical level, as a set of functional components and the information exchanged between them, so that it does not depend on particular products; the technologies used to implement it in this work are listed in Table 4.6.

The architecture was designed to meet four requirements. First, *bidirectional exchange*: schedules and commands flow from the coordination service to the equipment and interfaces, and state, events, and confirmations flow back to trigger new scheduling cycles. Second, *decoupling*: components communicate through an integration middleware with a publish/subscribe and request/response model, so that a component (e.g., an interface or a robot) can be added or replaced without changing the others. Third, *interoperability with industrial equipment*: the MMS controller is accessed through a standard industrial protocol rather than a proprietary interface. Fourth, *traceability*: agent, job, and equipment data, as well as execution logs, are kept in a persistent repository, which supports both operation and the experimental analysis.

Figure 4.15 shows the resulting logical architecture. The *coordination service* hosts the optimization modules of Chapter 5, a task manager that sequences robot skills [161], and a system monitor. It exchanges information with four groups of components through the integration middleware. The *robot controller* executes skills (arm motion, gripper actions, waiting for operator input or for a given time) and reports joint and gripper states. The *MMS controller*, reached through an industrial gateway, executes the transport commands of the MPCs and the battery swaps, and reports the current station, orientation, battery level, and waiting and working times of each MPC, as well as the number of spare batteries in the ABC. The *operator interfaces* (the AR HMD and the touch screen HMIs, the latter served by an HMI back-end) receive task instructions and media and return the operator's commands and confirmations. Finally, the *data repository* stores agent information (capabilities, execution times, availability), job descriptions (task instructions and auxiliary media), and equipment data (transport times and available stations). Through the interfaces, operators can add, remove, start, and complete operations, issue direct commands to the MPCs, monitor system status, and access multimedia content associated with the current task.

Table 4.6 lists how each function was implemented in this work. In this implementation, the coordination service, the integration middleware, the HMI back-end, and the data repository were co-located on a single server and run within a container to ensure that the environment can be reproduced. Other implementations of the same architecture are possible; for example, the integration middleware could be replaced by another message-oriented middleware and the solver by a commercial one, without changing the information flows in Figure 4.15.

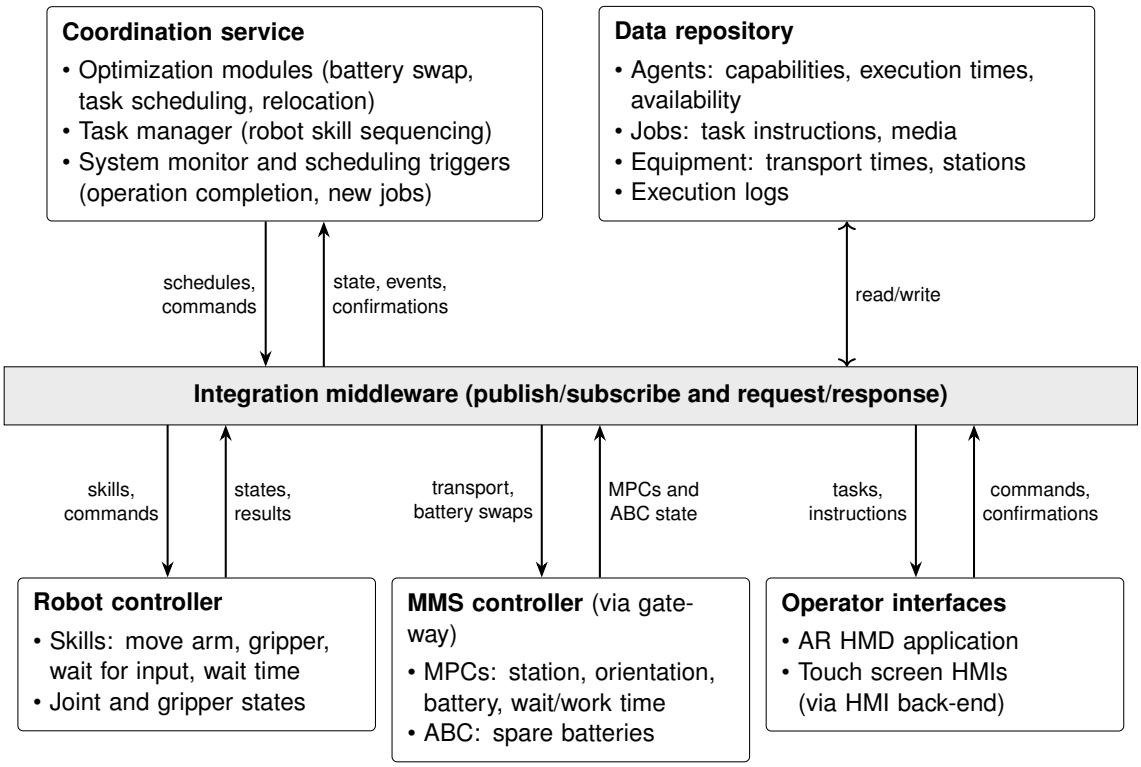


Figure 4.15: Logical architecture of the system: functional components and the information exchanged between them through the integration middleware. The technologies used to implement each component in this work are listed in Table 4.6.

Table 4.6: Implementation of the functional components of the architecture in this work.

Function	**Implementation in this work**
Server hardware	Workstation with Intel® Core™ Ultra 9 285K (24 cores, 5.7 GHz, 36 MB cache) and 64 GB DDR5 RAM; Ubuntu 24.04; Podman[a] container
Integration middleware	ROS Noetic[b] (topics, services, and actions)
Optimization modules	Python with PuLP and the COIN-OR Branch-and-Cut solver (CBC)[c]
Task manager	INESC TEC Task Manager [161]
Robot controller	UR5 with Robotiq 2F-140 gripper; *Universal_Robots_ROS_Driver*[d] and gripper skills over TCP/IP
Industrial gateway to the MMS	Open Platform Communications – Unified Architecture (OPC-UA) client (*ros_opcua*[e]) connected to the Prolynk PLC
HMI back-end	Node-RED[f] with ROS, OPC-UA, and *uibuilder* nodes; *rosbridge*[g]
Touch screen HMI hardware	Intel NUC 13 Pro (Intel® Core™ i5-1340P, 32 GB DDR4 RAM, 1 TB SSD, Windows 11 Pro) with an iiyama ProLite T1624MSC 10-point capacitive touchscreen
AR interface	Microsoft HL2; Unity with MRTK; *ROS-TCP Connector*[h]
Data repository	XAMPP[i] local database server

[a] `https://podman.io/`
[b] `https://wiki.ros.org/noetic/Installation`
[c] `https://coin-or.github.io/pulp/`
[d] `https://github.com/UniversalRobots/Universal_Robots_ROS_Driver`
[e] `https://github.com/iirob/ros_opcua_communication`
[f] `https://nodered.org/`
[g] `https://github.com/RobotWebTools/rosbridge_suite`
[h] `https://github.com/Unity-Technologies/Unity-Robotics-Hub`
[i] `https://www.apachefriends.org`

Chapter 5

A Robotic Coordination Framework for Human-Robot Teams in Matrix Manufacturing Systems

The central contribution of this chapter is a robotic coordination framework for HRTs operating in MMSs, integrating collaborative workcells, mobile robot transportation, and battery-aware execution within a coordinated decision pipeline. Unlike prior studies reviewed in Section 3.1, which address task allocation, robot relocation, or battery management as separate problems, the proposed framework coordinates these functions to support the coherent execution of human-only, robot-only, and collaborative operations in a reconfigurable manufacturing environment. This coordination is enabled by three MILP modules responsible for operation assignment and sequencing, mobile robot relocation, and battery swapping. The modules are solved sequentially and exchange key execution information, allowing the system to account for agent capabilities, workstation availability, transport times, and battery-state constraints while maintaining coherent system-level behavior. The framework therefore goes beyond a purely scheduling-oriented formulation, demonstrating how coordination logic can be embedded into a robotic manufacturing system with mobile and collaborative agents operating on a deployed industrial MMS. A video of the coordinated disassembly on the Prolynk platform is available at `https://youtu.be/iIhWYSqm_Z8`.

5.1 Problem Definition

Effective coordination of HRTs in MMSs represents a challenging problem due to its combinatorial structure, dynamic production conditions, resource limitations, and the need for coordinated interaction between human operators and robotic agents. Although a scheduling layer enhances system flexibility and adaptability, addressing this problem requires handling high-complexity constraints arising from multi-agent collaboration, variable operation processing times, transportation delays, and battery management. From this point forward, the term agent refers to both human

operators and industrial manipulators responsible for executing operations within the manufacturing system.

The FJSSP and the MMTSP exhibit structural similarities, as both consider a set of jobs composed of ordered sequences of operations to be processed by a set of machines. The key distinction lies in the execution modes. While the FJSSP assigns a single machine to each operation, the MMTSP allows multiple machines to execute an operation simultaneously, with each machine combination associated with a specific processing time. In both formulations, it is typically assumed that all jobs and machines are available at time zero, that each machine can process only one operation at a time, and that operations are non-preemptive (must be completed without interruption once started). The solution consists of assigning single or multiple machines to operations in order to optimize a selected objective function, most commonly the makespan.

This hybrid coordination problem integrates characteristics of both the FJSSP and the MMTSP, extended with mobile robot transportation. It consists of a set of j jobs $J = \{J_1, J_2, \ldots, J_j\}$, where each job J_j is composed of a sequence of o_j operations $O_j = \{O_{j,1}, O_{j,2}, \ldots, O_{j,o_j}\}$. The index $k \in \{1, 2, \ldots, o_j\}$ defines the execution order, which must be respected as in the FJSSP. Operations are performed at workstations by a set of agents $A = \{A_1, A_2, \ldots, A_a\}$, where an agent may be a human, a robot, or a HRT.

Each operation $O_{j,k}$ may be executed in multiple modes, corresponding to the different agents capable of performing it, with each mode A_i associated with a processing time $p_{j,k}^{(A_i)}$. This multi-mode, multi-agent structure is characteristic of the MMTSP formulation. Jobs are transported between workstations by a fleet of mobile robots $C = \{C_1, C_2, \ldots, C_c\}$, each capable of carrying at most one job at a time. Travel times between workstations are assumed known. Transportation is required only when two consecutive operations $O_{j,k}$ and $O_{j,k+1}$ of a job are assigned to different workstations; otherwise, no transport is necessary.

The objective is to determine feasible start times and agent assignments for all operations, along with transport allocations for the mobile robots, while satisfying precedence, resource, workstation, and transportation constraints and optimizing a selected performance criterion. As in both the FJSSP and the MMTSP, operations constitute the fundamental decision entities. Accordingly, the term task is used exclusively in the context of the scheduling algorithm, whereas the core elements of the problem are consistently referred to as operations.

5.2 Proposed Coordination Pipeline

To achieve efficient, adaptable, and effective coordination in MMS, three MILP modules are proposed to address task allocation and sequencing, mobile robot relocation, and battery swap scheduling, pursuing complementary but distinct objectives. The Battery Swap module runs first, determining which robots require charging and when, thereby establishing the energy availability constraints within which the remaining modules operate. The Task Scheduling module then performs the central allocation, assigning operations to HRTs and mobile robots with the overarching objective of minimizing makespan, while respecting agent capabilities, resource availability, and

the battery states inherited from the previous stage. The Mobile Robot Relocation module acts last, redistributing idle robots to unscheduled workstations to reduce spatial blocking and improve availability for subsequent scheduling cycles. The sequential execution of these modules is illustrated in Fig. 5.1, with their objectives, inputs, constraints, and outputs summarized in Table 5.1. This table shows the logic of the decision chain. The outputs of the Battery Swap module appear directly as inputs and feasibility constraints in the Task Scheduling module, whose own outputs, in turn, constrain the Mobile Robot Relocation module. This cascading structure is what allows three independently tractable subproblems to function as a coherent system-level coordination pipeline.

Table 5.1: Objectives, inputs, main constraints, and outputs of the three modules of the coordination pipeline. The outputs of each module are inputs of the following ones.

	Stage 1: Battery Swap	**Stage 2: Task Scheduling**	**Stage 3: Mobile Robot Relocation**
Objectives	1st: maximize total battery level of the MPCs; 2nd: minimize swap makespan; 3rd: minimize swap flowtime	1st: minimize makespan; 2nd: minimize flowtime	Minimize relocation makespan of idle MPCs
Inputs	MPCs requesting a battery swap; number and charge of the batteries in each ABC; travel times between stations and ABCs	Battery swap output; list of jobs and operations; available agents per station, with capabilities and execution times; travel times between stations and ABCs	Battery swap output; task scheduling output; travel times between stations and ABCs
Main constraints	ABC capacity (batteries on the rotating shaft); sequencing and no-overlap of MPCs and of slots; a slot inserts only if it holds a battery and retrieves only if empty; every retrieval is paired with an insertion	Precedence and non-preemption of operations; agent capabilities and execution times; task priorities; sequencing and no-overlap of MPCs and of stations; start times bounded by transport and swap completion	Each MPC assigned to exactly one station; at most one MPC per station; arrival times bounded by transport and swap completion
Outputs	Whether each MPC swaps its battery; ABC and slot pair (retrieval and insertion) for each swap; swap times (swap postponed if no slot holds a more charged battery)	Operation–agent assignment and execution mode; job–MPC and MPC–station allocation; start and end times	Idle MPC–station assignment

The design decision of decomposing the problem into three sequential modules rather than addressing it as a single monolithic formulation is motivated by the principle of separating concerns. The task scheduling, battery swap, and robot relocation problems are well-defined and independently tractable subproblems, each with established formulations in the literature, and treating

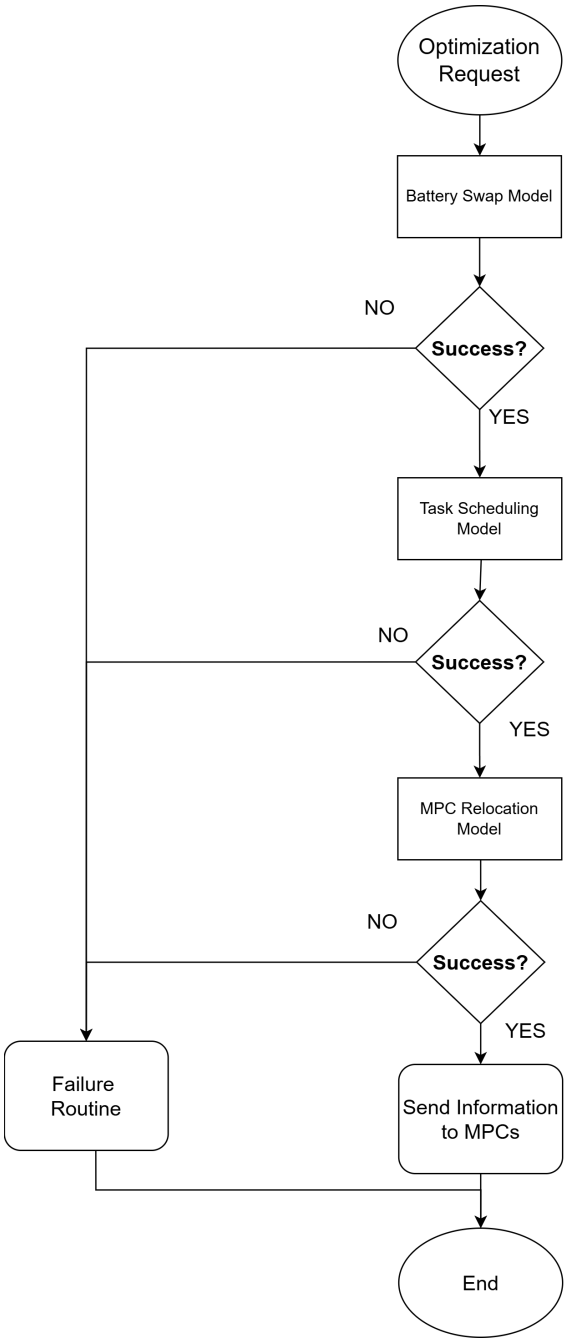


Figure 5.1: General flowchart of the coordination pipeline, illustrating the required inputs, main constraints, and outputs of each module.

them as such simplifies both implementation and adaptation to new scenarios. Necessary information is passed between modules at each stage, preserving system-level coherence without requiring a fully coupled formulation, lowering the overall complexity. The trade-off of this approach is the potential for suboptimality. Because each module optimizes locally given the outputs of the preceding one, globally optimal solutions are not guaranteed. In particular, since the Battery Swap module prioritizes replacements at the nearest available ABC station, the Task Scheduling module may assign robots to distant workstations, increasing transport times and producing schedules that a fully integrated formulation might improve upon.

The Battery Swap module runs first. It schedules the sequence of MPCs requiring a battery replacement and determines which ABC each mobile robot should visit. The module also selects the appropriate empty slots for battery retrieval and the occupied slots for battery replacement, aiming to maximize the battery levels of the robots undergoing a swap while minimizing process time. It accounts for the availability and charge levels of batteries in the ABC slots as well as the battery status of the robots. If no available battery provides a higher charge than the robot's current battery, the swap is postponed until a suitably charged battery becomes available.

The Task Scheduling module assigns each product to a MPC, workstation, agent, and the specific execution mode for the operation. This module accounts for transport times between workstations, the expected execution time of agents, the completion times of preceding operations, the time needed for battery replacement (if applicable), and the agents' capabilities for performing the assigned tasks.

The Mobile Robot Relocation module assigns idle MPCs to unscheduled workstations, freeing occupied workstations as needed. Similar to the Task Scheduling module, it accounts for transport times, battery replacement durations (if applicable), and workstation availability. To ensure up-to-date coordination, the Mobile Robot Relocation module runs immediately after the Task Scheduling module, allowing it to identify unscheduled workstations and optimize robot placement accordingly.

Several assumptions and constraints govern the coordination modules proposed in this work. Some are general properties of the problem class, while others arise specifically from the characteristics of the Prolynk MMS and are explicitly identified as such.

From a general problem standpoint, every operation is assumed to be feasible for at least one agent, a simplification adopted to avoid infeasible scheduling instances and ensure that the modules always produce a valid allocation. Battery replacement is treated as a reactive process. When a robot's battery falls below the threshold, it becomes eligible for replacement within the scheduling process and, if selected, is routed to the ABC. The replacement is performed immediately, with the time required incorporated into the next scheduled operation, rather than waiting for an optimal scheduling window. This modeling simplification trades scheduling flexibility for implementation tractability. Battery depletion is influenced by factors such as idle time, travel duration, and the frequency of acceleration and deceleration cycles, which makes precise timing nontrivial and justifies treating the threshold as a conservative safety margin rather than an exact value. If a new optimization request is submitted while a previous optimization is still in progress, the on-

going execution is aborted and the updated information is used to start a new scheduling cycle, a design choice motivated by the need to keep the coordination pipeline responsive to real-time changes in the MMS state.

Other constraints are specific to the Prolynk platform and its physical and operational characteristics. As described in Section 4.2, MPCs must always be assigned to a station, and only after indexing can they be reassigned to another station. This creates potential blocking situations. For example, if two MPCs are assigned to the same station, the one not yet indexed cannot move until the indexed robot vacates it. Critically, this also means that MPCs cannot be rerouted mid-transit, since rerouting is only possible when an MPC is indexed at a workstation or at the ABC. This physical behavior motivates two platform-specific modeling decisions. First, each workstation and the ABC can accommodate only one mobile robot at a time, and second, the number of mobile robots is constrained to be less than or equal to the number of available workstations. Both are architectural constraints imposed by the platform's indexing mechanism, not modeling simplifications. Similarly, the 20% battery swap threshold, with an additional 3% per robot to account for charging overhead, was empirically derived from the discharge and charge curves of the Prolynk MPCs measured under continuous operation, as detailed in Section 4.2, and is motivated by operational continuity. It guarantees that each robot can safely complete a battery replacement cycle before full depletion, ensuring at least 1.5 hours of continuous operation equivalent to a full recharge cycle.

5.3 Mathematical Formulations

For clarity, the sets and subscripts used throughout this section are defined in Tables 5.2 and 5.3. The decision variables and parameter notations used in the MILP modules are defined in Tables 5.4 and 5.5 for the Battery Swap module, Tables 5.6 and 5.7 for the Task Scheduling module, and Tables 5.8 and 5.9 for the Mobile Robot Relocation module.

It is important to note that some MILP formulations include multiple alternative objective functions, each solved in a separate run of the same module. The solution obtained from the first objective-function run is used as input for the second run, ensuring that the second stage builds upon the outcome of the first. Consequently, the Battery Swap module is executed three times and the Task Scheduling module twice. In the Battery Swap module, battery-level maximization is performed first, its resulting solution is used as input for the makespan minimization run, and the outcome of the latter is used as input for the flowtime minimization run. Similarly, Task Scheduling first minimizes makespan, after which the resulting schedule is used as input for the flowtime minimization run.

This sequential two-phase strategy was preferred over an explicit multi-objective formulation because it encodes a clear priority ordering, in which the second objective is optimized only within the solution space defined by the first. In the Battery Swap module, ensuring maximum battery availability takes precedence over minimizing makespan and flowtime, reflecting the fact that operational continuity is a precondition for any valid schedule; in the Task Scheduling module,

Table 5.2: Definition of subscripts used in the MILP formulations, specifying indices for agents, mobile robots, workstations, jobs, operations, battery changer slots, and null instances. This table provides a clear reference for interpreting variables and constraints in the coordination modules.

Indices	Description
a	Agent
e	Automatic battery changer
c	Mobile robot
j	Job
m	Workstation
m_c	Current workstation of mobile robot c
o	Operation
o_c	Current operation of mobile robot c
f_j	First operation of job j
l_j	Last operation of job j
s_l	Slot that retrieves the mobile robot battery
s_m	Slot that insert the mobile robot battery
s_e	Current slot of changer e
$s_\emptyset$, $s_{l\emptyset}$, $s_{m\emptyset}$, $c_\emptyset$, $e_\emptyset$	Represents a null instance of the element in the formulation.

Table 5.3: Definition of sets used in the MILP formulations, detailing groups of agents, mobile robots, workstations, jobs, operations, automatic battery changers, and slots. This table serves as a reference for understanding the components and structure of the coordination modules.

Set	Description
$\mathbb{A}$	Set of agents in workstation m
$\mathbb{E}$	Set of ABCs
$\mathbb{E}'$	$\mathbb{E}\backslash\{e_\emptyset\}$
$\mathbb{C}$	Set of mobile robots
$\mathbb{C}'$	$\mathbb{C}\backslash\{c_\emptyset\}$
$\mathbb{J}$	Set of jobs
$\mathbb{M}$	Set of workstations
$\mathbb{O}$	Set of operations
$\mathbb{S}$	Set of slots in ABC
$\mathbb{S}'$	$\mathbb{S}\backslash\{s_{m\emptyset}, s_{l\emptyset}\}$

makespan is minimized before flowtime, reflecting the system-level priority of completing all operations as early as possible before optimizing individual task latency. This structure guarantees that the primary objective is never sacrificed in favor of the secondary one, which would not be guaranteed by a weighted sum formulation.

Please note that *battery insertion* refers to placing a battery with a higher battery level from the ABC into the MPC, while *battery retrieval* refers to removing the depleted battery from the MPC and placing it into the ABC. A *battery swap* denotes the combined process of retrieving the depleted battery and inserting a charged one during the same visit. We use *battery change* or

battery replacement as general terms for the complete swap event.

5.3.1 Battery Swap Formulation

The Battery Swap module answers three questions: *which* mobile robot should swap its battery, at *which* ABC, and using *which* slots. Effective coordination of HRC systems must account for energy constraints, particularly when mobile robots play a crucial role in material transport. Battery swaps introduce additional coordination complexity, as they must be planned to minimize downtime while ensuring continuous operation. The proposed MILP module optimizes the sequencing of battery replacements, determining when and where a mobile robot should swap its battery based on current charge levels and available resources. The objective is to maximize overall battery levels across the system while minimizing delays and disruptions. By incorporating transport times, available charging slots, and the state of charge of each battery, this module ensures seamless integration of battery swaps within the broader coordination pipeline.

It is worth distinguishing between the general formulation and the platform-specific instantiation used in this work. In its general form, the module supports multiple ABCs and is not restricted to any particular battery swap threshold or occupancy rule. The specific constraints adopted here are not intrinsic to the module but reflect the physical architecture and operational characteristics of the Prolynk platform, as discussed in Section 4.2. This distinction is important since it clarifies that the module's applicability is not limited to the experimental setup used as experimental validation, and that relaxing or replacing these platform-specific constraints would not require reformulating the module from scratch, but only adjusting the corresponding parameters and feasibility conditions.

The variable and parameter structure of the Battery Swap module, summarized in Tables 5.4 and 5.5, reflects the combinatorial nature of the slot assignment problem. The decision variables fall into three functional groups. Assignment variables (L, I, K) that determine which slots retrieve and insert batteries for each robot; sequencing variables (P, U) that establish the order in which robots visit the ABC; and timing variables (R, T) that compute the start and end times of each swap operation. Together, these groups capture the three dimensions of the problem, defining what happens, in what order, and when. Their interaction is what makes the formulation non-trivial since the timing of each swap depends on the slot state left by the preceding robot, which is itself determined by the sequencing variables. The parameters provide the inputs that characterize the energy state of the system at the moment the module is triggered, and fall into two complementary groups. State parameters ($b_{e,s}$, b_c, q_e) that describe the current battery levels of both the robots and the ABC slots, and structural parameters ($t_{m_c,e}$, $t_{s,s'}$, t_{o_r}, t_{o_i}) that encode the time costs of traveling to the ABC and performing the physical retrieval and insertion operations. The distinction between these two groups reflects the reactive nature of the module. The state parameters change at every invocation, making the battery swap schedule sensitive to the current system state, while the structural parameters are fixed properties of the platform that remain constant across scheduling cycles.

Table 5.4: Definition of decision variables used in the Battery Swap module, describing the binary and continuous variables that represent battery presence, mobile robot sequencing, slot assignments, and the start and end times of battery swap operations. This table provides a reference for understanding the mathematical formulation of the Battery Swap module.

Decision Variable	Description
B_{total}	Total battery level
C_{max}	Makespan
C_{total}	Flowtime
$H_{e,c,c',s}$	Binary variable indicating whether slot s of changer e contains a battery immediately after the battery swap operation of mobile robot c, given that mobile robot c' performed its battery swap directly before mobile robot c at the same changer.
K_{e,c,s_l,s_m}	Binary variable that indicates whether the mobile robot c is going to change the battery at changer e using slots s_l and s_m
$L_{e,c,s}$	Binary variable indicating whether slot s of changer e retrieves the battery of the mobile robot c. If $e = e_\emptyset$, no battery is retrieved
$I_{e,c,s}$	Binary variable indicating whether slot s of changer e inserts the battery in the mobile robot c. If $e = e_\emptyset$, no battery is inserted
$P_{e,c,c'}$	Binary value indicating whether mobile robot c performs its battery swap immediately before mobile c' at changer e. If $c = c_\emptyset$, then c' is considered the first mobile robot to perform a swap at changer e, therefore it swaps battery before any other robot.
$Q_{e,c,s}$	Binary variable indicating the presence of battery in slot s of changer e after the mobile robot c makes its swap
R_{e,c,s_l,s_m}	Start time of battery swap of the mobile robot c in changer e using slot s_l to retrieve battery and slot s_m to insert battery
T_{e,c,s_l,s_m}	End time of battery swap of the mobile robot c in changer e using slot s_l to retrieve battery and slot s_m to insert battery
$U_{e,c,s_l,s_m,c',s'_l,s'_m}$	Binary variable indicating whether the mobile robot c swaps battery in changer e using slot s_l to retrieve battery and slot s_m to insert battery before the mobile robot c' using slot s'_l to retrieve battery and slot s'_m to insert battery

Table 5.5: Definition of parameters used in the Battery Swap module, describing battery levels, slot capacities, transport times, operation durations, and auxiliary constants. This table provides a reference for the mathematical representation of the Battery Swap module.

Parameters	Description
$b_{e,s}$	Battery level of the battery present in slot s of changer e
b_c	Mobile robots c current battery level
q_e	Number of batteries in changer e
t_{o_i}, t_{o_r}	Battery insert and retrieve time
t_{s_l,s_m}	Transport time between of previous slot s_l and slot s_m
$t_{m_c,e}$	Transport time between the mobile robot c current workstation m and changer e
M	A big number

Variable domains and index convention.

The decision variables of the Battery Swap module are defined on the following domains:

$$H_{e,c,c',s},\ I_{e,c,s},\ K_{e,c,s_l,s_m},\ L_{e,c,s},\ P_{e,c,c'},\ Q_{e,c,s},\ U_{e,c,s_l,s_m,c',s'_l,s'_m} \in \{0,1\},$$
$$R_{e,c,s_l,s_m},\ T_{e,c,s_l,s_m},\ B_{total},\ C_{max},\ C_{total} \geq 0.$$

To keep the constraints compact, quantifiers are omitted from the equations: unless otherwise indicated next to a constraint, each constraint holds for every value of its free indices in the default sets $e \in \mathbb{E}'$, $c, c' \in \mathbb{C}'$ with $c \neq c'$, and $s, s_l, s_m, s'_l, s'_m \in \mathbb{S}'$. Indices bound by a summation range only over the set indicated under the summation symbol, and null elements ($e_\emptyset$, $c_\emptyset$, $s_\emptyset$) appear only where they are written explicitly. M denotes a sufficiently large constant (big-M).

Equation (5.1a) and constraint (5.1b) define the objective function of maximizing the sum of the mobile robots' battery levels to evaluate the necessity of battery swaps. Equation (5.1c) and constraint (5.1d) define the objective function of minimizing the time required for battery swap operations, and Equation (5.1e) and constraint (5.1f) define the objective function of minimizing the flowtime of the battery swaps, i.e., the sum of their end times.

$$\max \quad B_{total} \tag{5.1a}$$
$$B_{total} = \sum_{c \in \mathbb{C}'} \left[\sum_{e \in \mathbb{E}', s_m \in \mathbb{S}'} (I_{e,c,s_m} \cdot b_{e,s_m} + I_{e_\emptyset,c,s_{m\emptyset}} \cdot b_c) \right] \tag{5.1b}$$
$$\min \quad C_{max} \tag{5.1c}$$
$$C_{max} \geq T_{e,c,s_l,s_m}, \qquad \forall e \in \mathbb{E},\ c \in \mathbb{C},\ s_l, s_m \in \mathbb{S} \tag{5.1d}$$
$$\min \quad C_{total} \tag{5.1e}$$
$$C_{total} = \sum_{e \in \mathbb{E}', c \in \mathbb{C}', s_l, s_m \in \mathbb{S}'} T_{e,c,s_l,s_m} \tag{5.1f}$$

The first group of constraints governs slot assignment feasibility, which robot uses which slot at which ABC. The key modelling challenge is that each slot can serve only one robot at a time, each robot can be assigned to only one slot, and the total number of swaps the ABC can accommodate is bounded by how many batteries it holds. Critically, this group also encodes the binary swap decision. By including the null ABC $e_\emptyset$ and null slot $s_\emptyset$ in the summation domain, a robot that does not swap is assigned to the null entity, eliminating the need for a separate indicator variable. Constraints (5.2a) and (5.2b) ensure that each ABC can accommodate a maximum number of mobile robots equal to the number of batteries available in the ABC. Constraints (5.2c) and (5.2d) ensure that each mobile robot can use only one slot to insert its battery and that each slot can only insert at most one battery, respectively. Similarly, (5.2e) and (5.2f) enforce the same constraints for retrieving slots. Constraints (5.2c) and (5.2e) also enforce a decision on whether or not a mobile robot must swap its battery.

$$\sum_{c\in\mathbb{C}',s\in\mathbb{S}'} I_{e,c,s} \leq q_e \tag{5.2a}$$

$$\sum_{c\in\mathbb{C}',s\in\mathbb{S}'} L_{e,c,s} \leq q_e \tag{5.2b}$$

$$\sum_{e\in\mathbb{E},s\in\mathbb{S}} I_{e,c,s} = 1 \tag{5.2c}$$

$$\sum_{c\in\mathbb{C}'} I_{e,c,s} \leq 1 \tag{5.2d}$$

$$\sum_{e\in\mathbb{E},s\in\mathbb{S}} L_{e,c,s} = 1 \tag{5.2e}$$

$$\sum_{c\in\mathbb{C}'} L_{e,c,s} \leq 1 \tag{5.2f}$$

The second group enforces battery conservation and the physical feasibility of each swap. A swap involves removing a depleted battery from the robot and inserting a charged one; the ABC must therefore have a charged battery available to insert and a free slot to receive the depleted one. These constraints prevent the module from assigning a slot to an operation it cannot physically perform given the current state of the ABC. Constraints (5.3a) and (5.3b) ensure that if a mobile robot is swapping its battery in ABC e, the ABC must retrieve and insert the battery, and the number of batteries leaving the ABC equals the number entering. Constraints (5.3c) and (5.3d) ensure that only slots containing a battery can perform the insertion operation, while (5.3e) and (5.3f) ensure that only empty slots can perform the retrieval operation.

$$\sum_{e\in\mathbb{E}',s\in\mathbb{S}'} I_{e,c,s} = \sum_{e\in\mathbb{E}',s\in\mathbb{S}'} L_{e,c,s} \tag{5.3a}$$

$$\sum_{c\in\mathbb{C}'} I_{e,c,s} = \sum_{c\in\mathbb{C}'} L_{e,c,s} \tag{5.3b}$$

$$I_{e,c,s} \leq Q_{e,c',s} + M\cdot(1-P_{e,c',c}) \tag{5.3c}$$

$$I_{e,c',s} \leq Q_{e,c,s} + M\cdot(1-P_{e,c,c'}) \tag{5.3d}$$

$$L_{e,c,s} \leq (1-Q_{e,c',s}) + M\cdot(1-P_{e,c',c}) \tag{5.3e}$$

$$L_{e,c',s} \leq (1-Q_{e,c,s}) + M\cdot(1-P_{e,c,c'}) \tag{5.3f}$$

The third group closes the remaining logical gaps in the slot-level assignment decisions. Even though the previous constraints correctly limit how many robots can use each slot, they do not prevent a robot from being assigned to the same slot for both retrieval and insertion, nor do they prevent the sequencing variable P from being active for a pair of robots that are not both visiting the same ABC. Constraints (5.4a) enforce that the retrieval and insertion slots must be different, for each robot. Constraint (5.4b) ensures that if a mobile robot does not select a retrieval slot, it must not select an insertion slot either, enforcing coherence between the two assignment decisions.

Constraints (5.4c) and (5.4d) bind the sequencing variable P to the assignment variables L and I, ensuring that a sequencing ordering between two robots can only be active if both robots are actually performing a swap at the same ABC.

$$L_{e,c,s} + I_{e,c,s} \leq 1 \tag{5.4a}$$

$$L_{e_\emptyset,c,s_\emptyset} = I_{e_\emptyset,c,s_\emptyset} \tag{5.4b}$$

$$P_{e,c,c'} \leq \frac{1}{4}\Big(\sum_{s\in\mathbb{S}'} L_{e,c,s} + \sum_{s\in\mathbb{S}'} I_{e,c,s} + \sum_{s'\in\mathbb{S}'} L_{e,c',s'} + \sum_{s'\in\mathbb{S}'} I_{e,c',s'}\Big) \tag{5.4c}$$

$$P_{e,c_\emptyset,c'} \leq \frac{1}{2}\Big(\sum_{s'\in\mathbb{S}'} L_{e,c',s'} + \sum_{s'\in\mathbb{S}'} I_{e,c',s'}\Big) \tag{5.4d}$$

The fourth group tracks the evolving state of the ABC slot by slot across successive swaps. This is necessary because the slot state after one robot's swap directly determines which slots are available for the next robot in the sequence, creating a chain of temporal dependencies that makes the battery swap formulation combinatorially complex. Constraint (5.5a) ensures that the total number of batteries in the ABC is conserved across each swap, one battery is removed and one is returned, leaving the total unchanged. Constraints (5.5b), (5.5c), and (5.5d) propagate this state forward by computing the battery presence in each slot immediately after each robot's swap, conditioning the result on which robot swapped immediately before and what the slot state was at that point.

$$\sum_{s\in\mathbb{S}'} Q_{e,c,s} = q_e \tag{5.5a}$$

$$H_{e,c,c',s} \leq Q_{e,c',s} + L_{e,c,s} - I_{e,c,s} + M\cdot(1 - P_{e,c',c}) \tag{5.5b}$$

$$H_{e,c',c,s} \leq Q_{e,c,s} + L_{e,c',s} - I_{e,c',s} + M\cdot P_{e,c,c'} \tag{5.5c}$$

$$Q_{e,c,s} = \sum_{c'\in\mathbb{C}':c\neq c'} H_{e,c,c',s} \tag{5.5d}$$

The fifth group introduces the timing dimension of the swap. Having decided which robot uses which slots, the module must compute when each swap takes place and ensure no robot can begin before physically arriving at the ABC. The variable K serves as a joint indicator that consolidates the two assignment decisions (L and I) into a single binary trigger for the timing constraints, active only when both the retrieval and insertion slots have been simultaneously assigned to the same robot at the same changer. Constraints (5.6a), (5.6b), and (5.6c) define variable K, activating it only when robot c is assigned to swap at changer e using both slots s_l and s_m simultaneously. Constraints (5.6d) and (5.6e) force the start and end times to zero when the mobile robot is not assigned to that specific swap, preventing spurious timing values. Constraint (5.6f) grounds the timing in physical reality, the swap cannot begin before the robot has had time to travel from its current workstation to ABC e.

$$K_{e,c,s_l,s_m} \leq L_{e,c,s_l} \tag{5.6a}$$
$$K_{e,c,s_l,s_m} \leq I_{e,c,s_m} \tag{5.6b}$$
$$K_{e,c,s_l,s_m} \geq L_{e,c,s_l} + I_{e,c,s_m} - 1 \tag{5.6c}$$
$$R_{e,c,s_l,s_m} \leq M \cdot K_{e,c,s_l,s_m} \tag{5.6d}$$
$$T_{e,c,s_l,s_m} \leq M \cdot K_{e,c,s_l,s_m} \tag{5.6e}$$
$$R_{e,c,s_l,s_m} \geq t_{m_c,e} - M \cdot (2 - L_{e,c,s_l} - I_{e,c,s_m}) \tag{5.6f}$$

The sixth group enforces no-overlap between swaps at the same ABC and ensures each swap lasts at least as long as the physical operations it comprises. Because the ABC rotating shaft can only serve one robot at a time, swaps must be sequenced without overlap, and the duration of each swap depends on which slots are involved since the shaft must travel between slot positions. Constraints (5.7a) and (5.7b) implement the no-overlap condition, the start time of each swap must exceed the end time of the immediately preceding swap at the same ABC. Constraints (5.7c) and (5.7d) enforce the minimum swap duration, which includes shaft travel from the previous slot position to the retrieval slot, the battery retrieval time, shaft travel to the insertion slot, and the battery insertion time. Constraint (5.7e) handles the special case where the swap has no immediately preceding operation at the ABC, using the shaft's current position as the starting reference.

$$R_{e,c,s_l,s_m} \geq T_{e,c',s'_l,s'_m} - M \cdot (4 - L_{e,c,s_l} - I_{e,c,s_m} - L_{e,c',s'_l} - I_{e,c',s'_m} + U_{e,c,s_l,s_m,c',s'_l,s'_m}) \tag{5.7a}$$
$$R_{e,c',s'_l,s'_m} \geq T_{e,c,s_l,s_m} - M \cdot (5 - L_{e,c,s_l} - I_{e,c,s_m} - L_{e,c',s'_l} - I_{e,c',s'_m} - U_{e,c,s_l,s_m,c',s'_l,s'_m}) \tag{5.7b}$$
$$\begin{aligned} T_{e,c,s_l,s_m} \geq\ & R_{e,c,s_l,s_m} + t_{s'_m,s_l} + t_{o_r} + t_{s_l,s_m} + t_{o_i} - M \cdot (4 - L_{e,c,s_l} - I_{e,c,s_m} \\ & - L_{e,c',s'_l} - I_{e,c',s'_m} + U_{e,c,s_l,s_m,c',s'_l,s'_m}) \end{aligned} \tag{5.7c}$$
$$\begin{aligned} T_{e,c',s'_l,s'_m} \geq\ & R_{e,c',s'_l,s'_m} + t_{s_m,s'_l} + t_{o_r} + t_{s'_l,s'_m} + t_{o_i} - M \cdot (5 - L_{e,c,s_l} - I_{e,c,s_m} \\ & - L_{e,c',s'_l} - I_{e,c',s'_m} - U_{e,c,s_l,s_m,c',s'_l,s'_m}) \end{aligned} \tag{5.7d}$$
$$T_{e,c,s_l,s_m} \geq R_{e,c,s_l,s_m} + t_{s_e,s_l} + t_{o_r} + t_{s_l,s_m} + t_{o_i} - M \cdot (3 - L_{e,c,s_l} - I_{e,c,s_m} - U_{e,c_\emptyset,s_\emptyset,s_\emptyset,c,s_l,s_m}) \tag{5.7e}$$

The seventh group completes the sequencing structure by binding the ordering variables to the assignment variables and ensuring the sequence of swaps at each ABC forms a well-defined chain with a unique starting point. Without these constraints, the solver could construct orderings between robots that are not both visiting the same ABC, producing a logically contradictory sequence. Constraints (5.8a) and (5.8b) bind variable U to the assignment variables L and I, ensuring that a slot-level ordering between two swap events can only be active if both events are actually assigned. Constraint (5.8c) ensures that at most one battery swap is designated as the first at each ABC, establishing a unique entry point for the ordering chain. Constraint (5.8d) enforces global consistency by requiring the total number of ordering relationships to match the total number of

swaps performed at each ABC. Constraints (5.8e) and (5.8f) bind the slot-level ordering variable U to the robot-level ordering variable P, ensuring both representations are consistent. Finally, constraint (5.8g) prevents circular orderings by ensuring that a swap designated as the first operation at a changer cannot simultaneously have a preceding robot.

$$U_{e,c,s_l,s_m,c',s'_l,s'_m} \leq \frac{1}{4}\left(L_{e,c,s_l} + I_{e,c,s_m} + L_{e,c',s'_l} + I_{e,c',s'_m}\right) \tag{5.8a}$$

$$U_{e,c_\emptyset,s_{l\emptyset},s_{m\emptyset},c',s'_l,s'_m} \leq \frac{1}{2}\left(L_{e,c',s'_l} + I_{e,c',s'_m}\right) \tag{5.8b}$$

$$\sum_{e\in\mathbb{E}',c'\in\mathbb{C}'} P_{e,c_\emptyset,c'} \leq |E| \tag{5.8c}$$

$$\sum_{c,c'\in\mathbb{C}':c\neq c'} P_{e,c,c'} = \sum_{c'\in\mathbb{C}',s\in\mathbb{S}'} I_{e,c',s} \tag{5.8d}$$

$$P_{e,c,c'} = \sum_{s_l,s_m,s'_l,s'_m\in\mathbb{S}'} U_{e,c,s_l,s_m,c',s'_l,s'_m} \tag{5.8e}$$

$$P_{e,c_\emptyset,c'} = \sum_{s'_l,s'_m\in\mathbb{S}'} U_{e,c_\emptyset,s_{l\emptyset},s_{m\emptyset},c',s'_l,s'_m} \tag{5.8f}$$

$$P_{e,c,c'} \leq 1 - P_{e,c_\emptyset,c'} \tag{5.8g}$$

5.3.2 Task Scheduling Formulation

The Task Scheduling module answers the questions *who* performs each operation, *where*, and *when*. Task scheduling in a MMS with HRTs presents a complex coordination challenge. Unlike traditional job-shop scheduling, where operations follow predefined routes, MMS allows dynamic task allocation, requiring an adaptive approach. Accordingly, the Task Scheduling module is the most complex and central component of the framework, comprising a richer set of decision variables that jointly handle operation assignment, sequencing, and timing across HRTs and mobile robots. Unlike the Battery Swap and Mobile Robot Relocation modules, which act reactively on a narrower set of decisions, this module must simultaneously reason about agent capabilities, operation precedences, resource availability, and the constraints imposed by battery swap operations, making it the primary locus of the coordination intelligence in the proposed framework. The objective is to minimize makespan and flowtime, supporting coherent execution of collaborative manufacturing operations between human operators and robotic agents.

The variable and parameter structure of the Task Scheduling module, summarized in Tables 5.6 and 5.7, reflects the hybrid nature of the problem, which simultaneously addresses allocation, sequencing, and timing across multiple agents and resources. The decision variables fall into three functional groups. The allocation variables (Y, W) that assign operations and jobs to agents, workstations, and mobile robots; sequencing variables (X, V, Z) that establish the ordering of jobs on mobile robots, operations performed by the same agent, and operations sharing a workstation; and timing variables (S, F) that compute the start and end times of each operation. This combination distinguishes the task scheduling formulation from a simple assignment problem, since the

feasibility of any allocation decision is conditioned on the sequencing constraints, which are in turn conditioned on the timing variables. The parameters reinforce this complexity by combining information from three distinct sources. Agent parameters (a_a, c_a, $p_{a,j,o,m}$) encode the heterogeneity of the HRT in terms of capabilities, availability, and execution times; priority parameters ($i_{a,j,o,m}$, $n_{a,j,o,m}$) govern the ordering in which operations are considered for allocation, reflecting task urgency and arrival order; structural parameters ($t_{m',m}$, $t_{e,m}$) couple scheduling decisions to the physical layout of the MMS through inter-workstation and ABC transport times; and inherited parameters (k_{e,c,s_l,s_m}, t_{e,c,s_l,s_m}), passed directly from the Battery Swap module, encode the energy constraints that the task schedule must respect. It is this last group that formally links the two modules and ensures that task scheduling decisions are always consistent with the battery state established in the preceding stage, making the interdependence between variables and parameters the defining characteristic of the formulation's complexity.

Table 5.6: Definition of decision variables used in the Task Scheduling module, describing operation sequencing, allocation of jobs to agents and mobile robots, and timing variables for start, end, and flow of tasks across workstations. This table serves as a reference for the mathematical formulation of the Task Scheduling module.

Decision Variables	Description
C_{max}	Makespan
C_{total}	Flowtime
$V_{a,j,o,m,j',o',m'}$	Binary variable indicating whether agent a performs operation o' of job j' in workstation m' after performing operation o of job j in workstation m
$F_{a,j,o,m}$	End time of operation o of job j on workstation$m$ performed by agent a
$S_{a,j,o,m}$	Start time of operation o of job j on workstation m performed by agent a
$X_{j,j',c}$	Binary variable indicating whether job j' is allocated after job j in the mobile robot c
$W_{j,c}$	Binary variable indicating whether job j is allocated to the mobile robot c
$Y_{a,j,o,m}$	Binary variable indicating whether operation o of job j is allocated to workstation m and performed by agent a
$Z_{a,j,o,a',j',o',m}$	Binary variable indicating whether operation o' of job j' performed by agent a' is allocated after operation o of job j performed by agent a in workstation m

Variable domains and index convention.

The decision variables of the Task Scheduling module are defined on the following domains:

$$V_{a,j,o,m,j',o',m'},\ W_{j,c},\ X_{j,j',c},\ Y_{a,j,o,m},\ Z_{a,j,o,a',j',o',m} \in \{0,1\}, \qquad S_{a,j,o,m},\ F_{a,j,o,m},\ C_{max},\ C_{total} \geq 0.$$

As in the Battery Swap module, quantifiers are omitted from the equations: unless otherwise indicated next to a constraint, each constraint holds for every value of its free indices in the default sets $a, a' \in \mathbb{A}$, $j, j' \in \mathbb{J}$, $o, o', o_c, o'_c \in \mathbb{O}$, $m, m' \in \mathbb{M}$, $c \in \mathbb{C}$, $e \in \mathbb{E}$, and $s_l, s_m \in \mathbb{S}$.

Table 5.7: Definition of parameters used in the Task Scheduling module, including agent availability and capabilities, operation priorities, execution and transport times, and values obtained from the Battery Swap module. This table provides a complete reference for the mathematical formulation of the Task Scheduling module.

Parameters	Description
a_a	Availability of agent a
c_a	Capability of agent a
$i_{a,j,o,m}$	Operations priority level of operation o of job j on workstation m performed by agent a
k_{e,c,s_l,s_m}	Binary value determined by the Battery Swap module K_{e,c,s_l,s_m}
$n_{a,j,o,m}$	Input order of operation o of job j on workstation m performed by agent a
$p_{a,j,o,m}$	Execution time of operation o of job j on workstation m performed by agent a
$t_{e,m}$	Transport time between ABC e and workstation m
t_{e,c,s_l,s_m}	Integer value determined by the Battery Swap module T_{e,c,s_l,s_m}
$t_{m',m}$	Transport time between workstations m' and m
M	A big number
$\delta_{a,j,o_c,m,\,a',j',o'_c,m'}$	Precomputed binary parameter equal to 1 if operation o_c of job j on agent a and station m has higher priority than o'_c of job j'; or, in case of equal priority, if it has a smaller First-In First-Out (FIFO) entry index

Equation (5.9a) and constraint (5.9b) define the objective function for minimizing the operations' makespan. Equation (5.9c) and constraint (5.9d) define the objective function for minimizing the operations' flowtime while ensuring that all operation end times are within the calculated makespan for flowtime computation.

$$\min \quad C_{max} \tag{5.9a}$$

$$C_{max} \geq F_{a,j,o,m} \tag{5.9b}$$

$$\min \quad C_{total} \tag{5.9c}$$

$$C_{total} = \sum_{a\in\mathbb{A},\, j\in\mathbb{J},\, o\in\mathbb{O},\, m\in\mathbb{M}} F_{ajom} \tag{5.9d}$$

The first group of task scheduling constraints establishes basic temporal feasibility. Each operation must last at least as long as its execution time, and each operation must start only after all logistical prerequisites have been met. This is where the inherited battery swap outputs are first explicitly used, since the swap completion time enters directly as a lower bound on the start time of the first operation of any job whose mobile robot has been scheduled for a battery replacement. Constraint (5.10a) ensures that each operation's end time is at least its start time plus its execution time. Constraint (5.10b) enforces that when a preceding operation exists within the same job, the next operation cannot begin before the preceding one finishes plus the inter-station travel time. Constraints (5.10c) and (5.10d) handle the case of a job's first operation. Without a preceding

operation, the start time is bounded by the robot's travel time from its current position and, when a battery swap has been scheduled, additionally by the swap completion time plus the travel time from the ABC to the target workstation. This directly links the Battery Swap module's output to the feasibility of the task schedule.

$$F_{a,j,o,m} \geq S_{a,j,o,m} + p_{a,j,o,m} - M \cdot (1 - Y_{a,j,o,m}) \tag{5.10a}$$

$$S_{a,j,o,m} \geq F_{a',j,o-1,m'} + t_{m',m} - M \cdot (2 - Y_{a',j,o-1,m'} - Y_{a,j,o,m}) \tag{5.10b}$$

$$S_{a,j,o,m} \geq t_{m',m} - M \cdot (2 - W_{j,c} - Y_{a,j,o,m}) \tag{5.10c}$$

$$S_{a,j,o,m} \geq t_{e,c,s_l,s_m} + t_{e,m} - M \cdot (3 - W_{j,c} - Y_{a,j,o,m} - k_{e,c,s_l,s_m}) \tag{5.10d}$$

The second group addresses the no-overlap requirements at two distinct levels, across jobs carried by the same mobile robot, and across operations sharing the same workstation or agent. The validity of any allocation decision depends entirely on the absence of resource conflicts in time, and the ordering variables X, Z, and V introduced here allow the module to reason about sequencing without fixing the order a priori. Constraints (5.11a) and (5.11b) enforce job sequencing on mobile robots. A robot cannot begin transporting a new job until it has finished delivering the previous one, accounting for inter-station travel time between consecutive deliveries. Constraints (5.11c) and (5.11d) regulate operation sequencing within each workstation, ensuring no two operations share the same station at the same time regardless of which agent performs them. Constraints (5.11e) and (5.11f) apply the same no-overlap logic to agents capable of operating at multiple stations, preventing a single agent from being simultaneously assigned to operations at different workstations.

$$S_{a,j,f_j,m} \geq F_{a',j',l'_j,m'} + t_{m',m} - M \cdot (5 - Y_{a',j',l'_j,m'} - Y_{a,j,f_j,m} - X_{j',j,c} - W_{j,c} - W_{j',c}), \qquad \forall\, j' < j \tag{5.11a}$$

$$S_{a',j',f'_j,m'} \geq F_{a,j,l_j,m} + t_{m,m'} - M \cdot (4 - Y_{a',j',f'_j,m'} - Y_{a,j,l_j,m} + X_{j,j',c} - W_{j,c} - W_{j',c}), \qquad \forall\, j' < j \tag{5.11b}$$

$$S_{a,j,o,m} \geq F_{a',j',o',m} - M \cdot (3 - Z_{a',j',o',a,j,o,m} - Y_{a,j,o,m} - Y_{a',j',o',m}), \qquad \forall\, j' < j \tag{5.11c}$$

$$S_{a',j',o',m} \geq F_{a,j,o,m} - M \cdot (2 + Z_{a',j',o',a,j,o,m} - Y_{a',j',o',m} - Y_{a,j,o,m}), \qquad \forall\, j' < j \tag{5.11d}$$

$$S_{a,j,o,m} \geq F_{a,j',o',m} - M \cdot (3 - V_{a,j',o',m',j,o,m} - Y_{a,j,o,m} - Y_{a,j',o',m}), \qquad \forall\, j' < j,\ m \neq m' \tag{5.11e}$$

$$S_{a',j',o',m} \geq F_{a,j,o,m} - M \cdot (2 + V_{a,j',o',m',j,o,m} - Y_{a',j',o',m} - Y_{a,j,o,m}), \qquad \forall\, j' < j,\ m \neq m' \tag{5.11f}$$

The third group enforces the structural integrity of the allocation decisions by linking ordering and sequencing variables to the allocation variables, and by imposing uniqueness of assignment. Without these constraints, the solver could construct orderings between operations or jobs that are not assigned to the same resource, producing a schedule that appears feasible mathematically but is logically contradictory. Constraints (5.12a), (5.12b), and (5.12c) bind the ordering variables V,

Z, and X to the allocation variables Y and W, ensuring that an ordering between two operations or jobs can only be active if both are actually assigned to the relevant resource. Constraint (5.12d) ensures that each job is carried by exactly one mobile robot, reflecting the physical constraint that each workpiece travels on a single transport unit. Constraint (5.12e) ensures that each operation is executed by exactly one agent at one workstation, preventing duplicate assignments.

$$V_{a,j',o',m',j,o,m} \leq \frac{1}{2}\Big(Y_{a,j,o,m} + Y_{a,j',o',m'}\Big) \tag{5.12a}$$

$$Z_{a,j,o,a',j',o',m} \leq \frac{1}{2}\Big(Y_{a,j,o,m} + Y_{a',j',o',m}\Big) \tag{5.12b}$$

$$X_{j,j',c} \leq \frac{1}{2}\Big(W_{j,c} + W_{j',c}\Big) \tag{5.12c}$$

$$\sum_{c\in\mathbb{C}} W_{j,c} = 1 \tag{5.12d}$$

$$\sum_{a\in\mathbb{A}}\sum_{m\in\mathbb{M}} Y_{a,j,o,m} = 1 \tag{5.12e}$$

The fourth group introduces the capability and priority dimensions, transforming the module from a generic scheduling problem into a human-robot coordination problem. Beyond deciding when and where operations take place, the module must ensure that each operation is assigned only to an agent that is both available and capable of performing it, and that when two operations compete for the same resource, the higher-priority one is scheduled first. Constraint (5.13a) filters the feasible allocation space. Operation o can only be assigned to agent a if a is currently available ($a_a = 1$) and capable of performing it ($c_a = 1$). Constraints (5.13b) and (5.13c) force start and end times to zero for agent-operation-workstation combinations that are not selected, preventing spurious non-zero timing values from appearing in the solution. Constraints (5.13d) and (5.13e) regulate execution order through the precomputed binary parameter δ. When two operations have different priority levels, the higher-priority one must start first; when they share the same priority, a FIFO rule applies, scheduling the operation that entered the system earlier before the one that entered later. This mechanism allows the module to respect urgency differences between jobs without requiring explicit enumeration of precedence pairs.

$$Y_{a,j,o,m} \leq a_a \cdot c_a \tag{5.13a}$$

$$S_{a,j,o,m} \leq M \cdot Y_{a,j,o,m} \tag{5.13b}$$

$$F_{a,j,o,m} \leq M \cdot Y_{a,j,o,m} \tag{5.13c}$$

$$S_{a,j,o_c,m} \leq S_{a',j',o'_c,m'} + M \cdot \big(3 - Y_{a,j,o_c,m} - Y_{a',j',o'_c,m'} - \delta_{a,j,o_c,m,a',j',o'_c,m'}\big) \tag{5.13d}$$

$$S_{a',j',o'_c,m'} \leq S_{a,j,o_c,m} + M \cdot \big(2 - Y_{a,j,o_c,m} - Y_{a',j',o'_c,m'} + \delta_{a,j,o_c,m,a',j',o'_c,m'}\big) \tag{5.13e}$$

Situating this formulation within the literature reviewed in Chapter 3, the Task Scheduling module extends both the classical FJSSP and the MMTSP in several important respects. From the

FJSSP, it inherits the operation sequencing structure and the no-overlap constraints that prevent simultaneous resource use; from the MMTSP, it adopts the multimode execution structure, where each operation may be performed by a human, a robot, or a HRT, each with a distinct processing time. However, as identified in Section 3.1.2, neither of these classical formulations accounts for the system-level logistics layer that emerges when multiple HRT stations must be coordinated within a larger production environment. The proposed module addresses this gap by explicitly incorporating inter-station transportation times, which couple the start time of each operation to the arrival time of the mobile robot carrying the workpiece from the preceding station. Furthermore, agent availability parameters (a_a) and capability parameters (c_a) encode the heterogeneity of the HRT workforce in a way that purely machine-centered FJSSP formulations do not, while the inherited battery swap parameters (k_{e,c,s_l,s_m}, t_{e,c,s_l,s_m}) introduce energy-driven temporal dependencies that are absent from all works reviewed in Section 3.1. Together, these extensions place the Task Scheduling module at the intersection of FJSSP, MMTSP, and mobile robot logistics, reflecting the multi-layered coordination challenge that arises in a realistic reconfigurable MMS.

5.3.3 Mobile Robot Relocation Formulation

The Mobile Robot Relocation module answers the question of *where* idle mobile robots should go so that the workstations needed next are free. In dynamic manufacturing environments, efficient utilization of workstations is crucial for maintaining productivity. Idle mobile robots can occupy valuable space between operations, potentially blocking other operations and creating bottlenecks. The Mobile Robot Relocation module addresses this challenge by optimizing the reassignment of idle robots to unoccupied workstations, ensuring that high-priority operations proceed without unnecessary delays. Importantly, workstation availability is only known after the Task Scheduling module has determined the operation assignments and timing. The Mobile Robot Relocation module accounts for transport times, workstation availabilities, and battery-swap constraints to determine an optimal reassignment strategy. By efficiently reallocating idle robots based on the finalized schedule, this approach enhances workstation accessibility, reduces congestion, and improves overall system flow.

It should be noted that minimizing relocation makespan is a local objective that does not directly optimize system-wide performance metrics such as overall makespan or throughput. By minimizing only the time required to complete all relocation movements, the module ensures that robots reach their assigned workstations as early as possible, which is a necessary but not sufficient condition for global efficiency. In principle, a relocation assignment that is locally optimal may not be the one that best positions robots for the subsequent scheduling cycle, since the module has no visibility into future task assignments.

The variable and parameter structure of the Mobile Robot Relocation module, summarized in Tables 5.8 and 5.9, reflects the auxiliary and reactive role of the module within the framework. With allocation and sequencing decisions already fixed by the Task Scheduling module, the relocation problem reduces to a straightforward assignment of remaining robots to remaining stations, requiring only three decision variables. The binary allocation variable (G) that assigns each idle

robot to a workstation, the continuous variable (D) that computes the resulting arrival time, and the makespan (C_{max}) that the module minimizes. The parameters are almost entirely inherited from the two preceding modules. The battery swap outputs (k_{e,c,s_l,s_m}, t_{e,c,s_l,s_m}) determine whether a robot must first complete a battery swap before being relocated, directly affecting its earliest possible arrival time at any workstation, while the transport times ($t_{m',m}$, $t_{e,m}$) determine the travel cost of each possible assignment. This parameter structure reflects the downstream position of the Mobile Robot Relocation module in the decision chain, where its role is not to re-solve the scheduling problem but to optimally place the remaining robots given what has already been decided.

Table 5.8: Definition of decision variables used in the Mobile Robot Relocation module, including makespan, travel times, and allocation indicators for mobile robots to workstations. This table summarizes all variables required to formulate the Mobile Robot Relocation module.

Decision Variables	Description
C_{max}	Makespan
$D_{m,c}$	End Travel time of the mobile robot c to workstation m if the mobile robot c is assigned to workstation m
$G_{m,c}$	Binary variable that indicates whether the mobile robot c is allocated to workstation m

Table 5.9: Definition of parameters used in the Mobile Robot Relocation module, including inputs derived from the Battery Swap module, inter-workstation travel times, and a large constant for modeling purposes. This table summarizes all numerical and binary parameters required for formulating the Mobile Robot Relocation module.

Parameters	Description
k_{e,c,s_l,s_m}	Binary value determined by the Battery Swap module K_{e,c,s_l,s_m}
t_{e,c,s_l,s_m}	Integer value determined by the Battery Swap module T_{e,c,s_l,s_m}
$t_{m',m}$	Transport time between workstations m' and m
$t_{e,m}$	Transport time between changer e and workstation m
M	A big number

Variable domains and index convention.

The decision variables of the Mobile Robot Relocation module are defined on the following domains:

$$G_{m,c} \in \{0,1\}, \qquad D_{m,c},\, C_{max} \geq 0.$$

Unless otherwise indicated, each constraint holds for every value of its free indices in the default sets $m \in \mathbb{M}$, $c \in \mathbb{C}$, $e \in \mathbb{E}$, and $s_l, s_m \in \mathbb{S}$.

Equation (5.14a) and constraint (5.14b) define the objective function of minimizing the relocation time.

$$\min \quad C_{max} \tag{5.14a}$$

$$C_{max} \geq D_{m,c} \tag{5.14b}$$

Constraint (5.15a) ensures that the end travel time of the mobile robot c to workstation m is non-zero if and only if the mobile robot c is allocated to workstation m. Constraints (5.15b) and (5.15c) calculate the end travel time of the mobile robot c to workstation m when a battery swap is requested and when it is not, respectively.

$$D_{m,c} \leq M \cdot G_{m,c} \tag{5.15a}$$

$$D_{m,c} \geq t_{m_c,m} - M \cdot (1 - G_{m,c}) \tag{5.15b}$$

$$D_{m,c} \geq t_{e,c,s_l,s_m} + t_{e,m} - M \cdot (2 - G_{m,c} - k_{e,c,s_l,s_m}) \tag{5.15c}$$

Finally, Constraints (5.16a) and (5.16b) ensure that each mobile robot can be allocated to only one workstation, and each workstation can receive only one mobile robot, respectively.

$$\sum_{m \in \mathbb{M}} G_{m,c} = 1 \tag{5.16a}$$

$$\sum_{c \in \mathbb{C}} G_{m,c} \leq 1 \tag{5.16b}$$

Taken together, these constraints serve three complementary purposes. Constraints 5.15a to 5.15c ensure temporal feasibility by guaranteeing that each robot's arrival time at its assigned workstation accounts for its current position and, if applicable, the time required to complete a battery swap before departing. Constraints 5.16a and 5.16b enforce exclusivity of occupation, ensuring that each robot is assigned to exactly one workstation and each workstation receives at most one robot. Collectively, these two groups ensure that the relocation decisions are both physically executable and conflict-free, while remaining consistent with the battery swap schedule inherited from the preceding module.

The Mobile Robot Relocation module was motivated by a specific constraint of the Prolynk platform. Since MPCs must always be assigned to a station, and since the platform prevents the rerouting of MPCs outside of indexing points, blocking situations may arise. The Mobile Robot Relocation module is therefore introduced to mitigate these scenarios proactively. While this requirement is partly driven by the platform's architecture, the underlying problem of assigning idle mobile carriers to available workstations while minimizing travel time and respecting occupancy constraints is broadly applicable to any MMS where repositioning resources between scheduling cycles can reduce transport delays and improve overall system availability.

5.4 Coordination Framework Experimental Evaluation

5.4.1 Use Case

To demonstrate the applicability of the proposed coordination framework, an experimental validation study was conducted using the Prolynk MMS. The use case focuses on the disassembly of a 3D-printed car alternator[13], selected for its combination of human-only, robot-only, and collaborative operations. This scenario enables validation of the framework's ability to coordinate human-robot teams, manage transportation and energy constraints, and support realistic production workflows in a deployed industrial environment.

The 3D-printed car alternator considered in this study is presented in Fig. 5.2. Its disassembly comprises nine operations performed by human (H), robot (R), or collaborative (C) agents, as detailed in Table 5.10. Transportation times between workstations and ABC slots, reported in Table 5.11 and Table 5.12, respectively, were obtained from empirical measurements within the MMS.

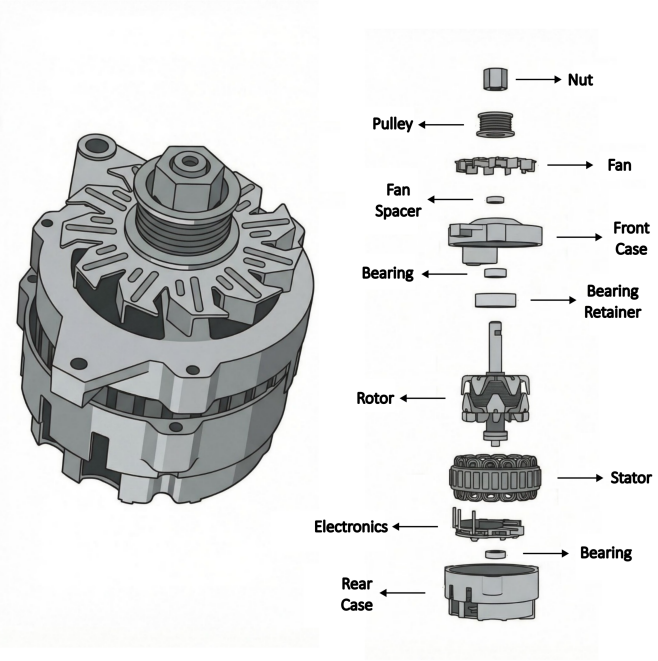


Figure 5.2: 3D-printed car alternator employed as the physical workpiece in the disassembly case study, representing a multi-component assembly used to evaluate task execution, operator guidance, and human-robot interaction within the proposed system.

As shown in Fig. 5.3, Operator 1 and Robot 1 were initially stationed at WS 10, while Operator 2 was assigned to WS 30. MPC 1, 2, and 3 were positioned at WS 10, WS 30, and WS 40, with battery levels of 95%, 68%, and 28%, respectively. These differentiated charge levels enabled evaluation of the framework's capability to determine which MPCs required battery replacement.

[13] https://www.thingiverse.com/thing:2681504

For demonstration purposes, the battery-change flag was activated for all MPCs. A single disassembly request was then submitted, assuming the product was located at the first workstation assigned to the selected MPC, thereby defining consistent initial conditions for the scheduling process. This configuration allowed assessment of the framework's effectiveness in allocating operations, coordinating heterogeneous resources, and minimizing makespan and flowtime under realistic operational constraints.

Table 5.10: Operations, components, and admissible agents of the car alternator disassembly case study, listed in disassembly order. H: human; H/R: human or robot; H/C: human or human-robot collaboration.

Op.	Components	Agents	Description
O0	Large nut	H	Remove the large fixing nut from the alternator shaft
O1	Pulley	H/R	Loosen and remove the pulley assembly
O2	Fan	H/R	Remove the cooling fan from the rotor shaft
O3	Case screws and nuts	H	Loosen all external case screws and nuts
O4	Front case, fan spacer, front case bearing, and bearing retainer	H/C	Remove the front case components
O5	Rotor	H/R	Extract the rotor from the stator assembly
O6	Stator	H/R	Remove the stator from the rear case
O7	Electronics	H	Disconnect and remove the electronic components
O8	Rear case and rear case bearing	H/C	Disassemble the rear case and its bearing

Table 5.11: Transportation times (in seconds) between workstations and the ABC, obtained from empirical observations, used for routing and scheduling of mobile robots.

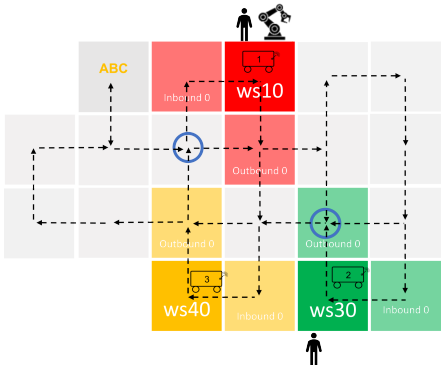
Workstation	Time (s)			
	WS 10	WS 30	WS 40	ABC
WS 10	0	29	25	34
WS 30	20	0	17	26
WS 40	12	32	0	23
ABC	16	30	27	0

5.4.2 Coordination Framework Experimental Evaluation Results

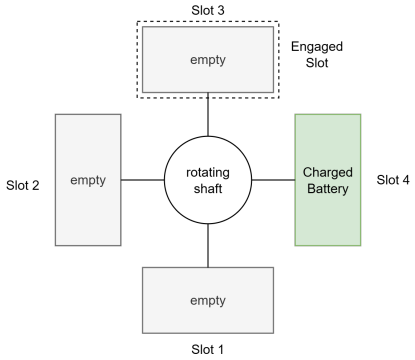
The resulting disassembly schedule of the first scheduling cycle is summarized in Table 5.13 and illustrated as a Gantt chart in Fig. 5.4. Operator 1 was assigned to operations 1 and 7, Robot 1 to operations 2 and 5, while both jointly executed the collaborative operations 4 and 8. Operator 2 was responsible for operations 0, 3, and 6. This allocation demonstrates the framework's capacity

Table 5.12: Transportation times (in seconds) between slots within the ABC, used for modeling battery swap operations and scheduling constraints.

Slot	**Time (s)**			
	Slot 1	Slot 2	Slot 3	Slot 4
Slot 1	0	1	2	3
Slot 2	1	0	1	2
Slot 3	2	1	0	1
Slot 4	3	2	1	0



(a) Agents and MPCs Distribution in the Prolynk MMS



(b) ABC Battery Distribution

Figure 5.3: Experimental setup of the Prolynk MMS: (a) spatial distribution of human operators, robotic agents, and MPCs across the system, illustrating task allocation and resource positioning; (b) battery disposition within the ABC.

to distribute workloads across human-only, robot-only, and collaborative operations while respecting precedence and capability constraints.

Table 5.13: Task allocation and corresponding execution times (in seconds) for each agent in the Car Alternator disassembly case study — First Scheduling Cycle.

Agent Index	**Agents**	**Workstation**	**Execution Time (s)**									**Allocated Operations**
			O0	O1	O2	O3	O4	O5	O6	O7	O8	
A1	Operator 1	WS10	70	4	25	153	41	39	72	5	68	1, 7
A2	Robot 1	WS10	–	20	20	–	–	22	63	–	–	2, 5
A3	Operator 1 + Robot 1	WS10	–	–	–	–	31	–	–	–	31	4, 8
A4	Operator 2	WS30	3	43	50	104	80	39	5	31	67	0, 3, 6
Operation End Times	-	-	3	27	47	180	231	253	287	312	343	-

Battery optimization selected MPC 3 for replacement, increasing its charge level from 28% to 100%. The module retrieved the depleted battery with slot 3 (pre-set as ready at initialization) and replaced it with the charged battery stored in slot 4 (the only slot containing a battery), with the overall swap process estimated at 32 seconds. Since the operation O0 was assigned to MPC 2, already located at WS 30, the Mobile Robot Relocation module reassigned MPC 1 to WS 40, allowing MPC 3 to move to WS 10, thereby reducing the overall relocation makespan. Computation



Figure 5.4: Gantt chart of the Task Scheduling module output for the first scheduling cycle. Each row corresponds to an agent and cycle pair, with bars coloured by agent type: Operator 1 (dark red), Robot 1 (blue), Op1+Robot1 collaborative (purple), and Operator 2 (green). Solid bars represent Cycle 1 assignments; lighter bars represent the projected Cycle 2 assignments computed in the same solve. The yellow vertical line marks the Cycle 2 trigger point, and the red vertical line indicates the Cycle 1 makespan (343 s).

times for this initial scheduling cycle are summarized in Table 5.14. The most computationally demanding module was task scheduling (34.74 s), followed by battery optimization (0.41 s) and robot relocation (0.01 s). The gap of 1.67 s between the total optimization time and the Python-measured solver time results from problem-construction overhead, which scales with the number of mobile robots, jobs, and agents.

Table 5.14: Performance of the First Scheduling Cycle: Computation Times and Optimal Values for Battery, Task Scheduling, and Mobile Robot Relocation modules.

Optimization	**Time (s)**	**Optimal Value**
Battery Level Optimization	0.32	263%
Battery Change Makespan Optimization	0.05	32s
Battery Change Flowtime Optimization	0.04	32s
Task Scheduling Makespan Optimization	22.18	343s
Task Scheduling Flowtime Optimization	12.56	1683s
Relocation Makespan Optimization	0.01	48s
Total Optimization Time	35.16	-
Python-measured Solver Time	36.83	-

A notable coordination behavior emerged during execution. When operation 0 at WS 30 completed, MPC 3 was still in transit to the ABC, which triggered a new scheduling cycle with the updated system state. Notably, the battery swap was not triggered for MPC 1 and MPC 2, as their battery levels remained above the 20% threshold. The resulting disassembly schedule of the second scheduling cycle is summarized in Table 5.15, and maintained the same agent and task

allocation as the first cycle, with only the MPC routing updated to reflect the current system state.

The new schedule assigned MPC 2 to WS 10 for operation 1, redirected MPC 3 to WS 40, and relocated MPC 1 to WS 30. This decision reflects the Mobile Robot Relocation module accounting for battery swap timing. Routing MPC 3 to WS 30 after the swap would cost 30 seconds of travel plus a minimum of 9 seconds for battery retrieval and release, totalling 39 seconds, whereas redirecting it to WS 40 would cost only 27 seconds of travel plus a minimum of 9 seconds, totalling 36 seconds. Since relocating MPC 1 to WS 30 costs only 32 seconds, the framework chose to relocate MPC 1 to WS 30 instead, saving 3 seconds in relocation makespan. Since MPC 3 had not yet arrived at the ABC when the new schedule was produced, it was indexed into the ABC, completed the battery swap, and was released directly towards WS 40 without any unnecessary detour. Had the scheduling cycle been triggered after MPC 3 had already departed the ABC, it would have continued towards its previously assigned station and only been rerouted upon receiving the updated command at the next indexing point, potentially increasing travel time. This illustrates a key property of the coordination pipeline. By triggering a new scheduling cycle at each operation completion and passing updated system state between modules, the framework makes positioning decisions that account for the ongoing physical state of the robots, exploiting timing windows that would be lost in a purely offline scheduling approach. Computation times for this second scheduling cycle are reported in Table 5.16.

Table 5.15: Task Allocation and Execution Times in the Car Alternator Disassembly Case Study — Second Scheduling Cycle. The ✓indicates operations already completed by the assigned agent before this scheduling cycle.

Agent Index	**Agents**	**Workstation**	**Execution Time (s)**									**Allocated Operations**
			O0	O1	O2	O3	O4	O5	O6	O7	O8	
A1	Operator 1	WS10	–	4	25	153	41	39	72	5	68	1, 7
A2	Robot 1	WS10	–	20	20	–	–	22	63	–	–	2, 5
A3	Operator 1 + Robot 1	WS10	–	–	–	–	31	–	–	–	31	4, 8
A4	Operator 2	WS30	✓	43	50	104	80	39	5	31	67	3, 6
Operation End Times	-	-	✓	24	44	177	228	250	284	309	340	-

Table 5.16: Performance of the Second Scheduling Cycle: Computation Times and Optimal Values for Battery, Task Scheduling, and Mobile Robot Relocation modules.

Optimization	**Time (s)**	**Optimal Value**
Battery Level Optimization	0.01	100%
Battery Change Makespan Optimization	0.01	9s
Battery Change Flowtime Optimization	0.01	9s
Task Scheduling Makespan Optimization	29.15	340s
Task Scheduling Flowtime Optimization	3.17	1656s
Relocation Makespan Optimization	0.19	36s
Total Optimization Time	32.54	-
Python-measured Solver Time	35.94	-

The coordination framework was successfully executed on the physical Prolynk platform without interruption. Across the disassembly task, the three MPCs performed a total of fifteen station transitions across ten scheduling cycles, one per operation completion, as summarized in Table 5.17 and visualized in Fig. 5.5. MPC 2, the only MPC carrying a job, performed five transitions between WS 30 and WS 10 to support the interleaved sequence of collaborative and manual operations. It travelled to WS 10 for operations 1 and 2, returned to WS 30 for operation 3, moved back to WS 10 for operations 4 and 5, returned to WS 30 for operation 6, and concluded at WS 10 for operations 7 and 8. MPC 1 and MPC 3, carrying no jobs, performed the remaining ten relocation transitions to free the required workstations ahead of MPC 2's arrivals, including MPC 3's initial trip from WS 40 to the ABC for battery replacement before rejoining the workflow. Each operation completion triggered a scheduling cycle with an updated system state, computing the next assignment and issuing relocation commands to clear the next required workstation. At the end of execution, MPC 1 remained at WS 30, MPC 2 at WS 10, and MPC 3 at WS 40.

Table 5.17: MPC Station Assignments Across Scheduling Cycles. Each scheduling cycle is triggered by the completion of the indicated operation, except for cycle 1 which corresponds to the initial scheduling request. The last row (End) reflects the final system state after operation 8 completes. (*) indicates a destination that was rescheduled in the following cycle before the MPC reached it.

Scheduling Cycle	**Operation**	**MPC 1**	**MPC 2**	**MPC 3**
Start (1)	-	WS10	WS30	WS40
2	O0	WS40	WS30	ABC ($\rightarrow$ WS10)*
3	O1	WS30	WS10	ABC $\rightarrow$ WS40
4	O2	WS30	WS10	WS40
5	O3	WS40	WS30	WS10
6	O4	WS40	WS10	WS30
7	O5	WS40	WS10	WS30
8	O6	WS10	WS30	WS40
9	O7	WS30	WS10	WS40
10	O8	WS30	WS10	WS40
End	-	WS30	WS10	WS40

5.4.3 Discussion and Conclusions of the Coordination Framework

The case study supports the feasibility and practical applicability of the proposed coordination framework for HRTs in a reconfigurable MMS. The Task Scheduling module successfully distributed the disassembly workload across heterogeneous agents, supporting coherent execution of human-only, robot-only, and collaborative operations in a manner consistent with precedence constraints and agent capabilities, achieving a makespan of 343 s and a flowtime of 1683 s in the first scheduling cycle. The Battery Swap module correctly identified MPC 3 as requiring replacement and determined an optimal slot assignment that maximized battery level, increasing the charge of MPC 3 from 28% to 100% within an estimated swap time of 32 s. The Mobile Robot Relocation module subsequently repositioned idle MPCs to free the required workstations ahead of MPC 2's

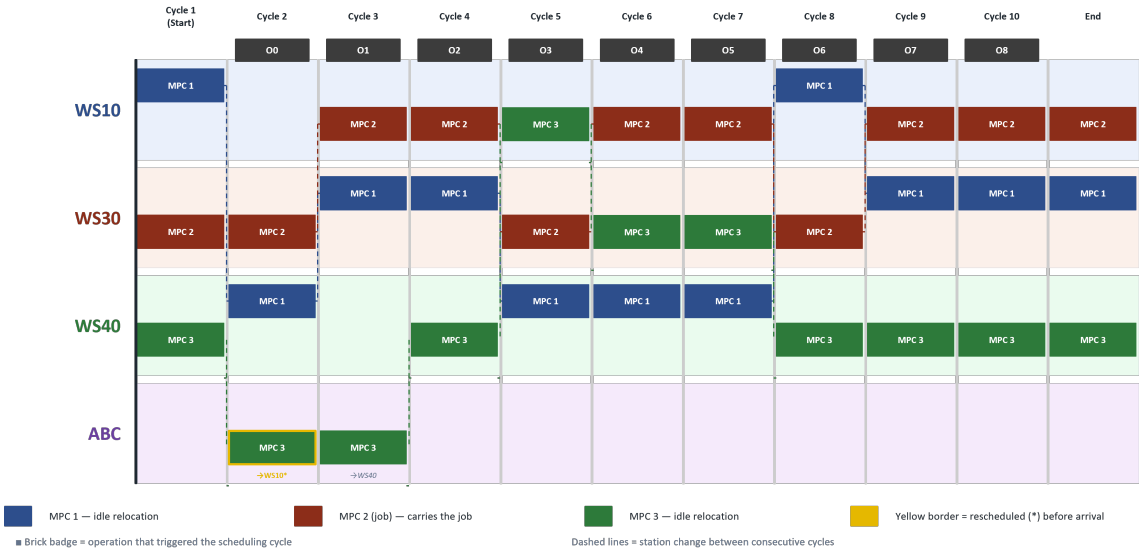


Figure 5.5: MPC station assignments across all ten scheduling cycles of the car alternator disassembly case study. Each row corresponds to a workstation (WS10, WS30, WS40, ABC), and each column to a scheduling cycle triggered by the completion of the indicated operation. Blocks are coloured by MPC role: MPC 1 idle relocation (blue), MPC 2 job carrier (dark red), and MPC 3 idle relocation (green). Dashed vertical lines denote station changes between consecutive cycles; a yellow border marks a destination that was rescheduled before the MPC reached it (*).

arrivals, with a relocation makespan of 48 s in the first cycle. Taken together, the three modules produced a coherent and conflict-free coordination plan, confirming that the sequential pipeline is capable of propagating decisions across stages without loss of system-level consistency.

Table 5.18 compares the proposed framework with the closest works reviewed in Section 3.1. The HRT scheduling approaches address the allocation of operations between humans and robots, several of them with physical validation and, in some cases, reactive rescheduling, but they consider a single workcell or station without transport between stations. The FJSSP formulations that include transport resources consider machines rather than human-robot teams and are validated on benchmark instances. None of the reviewed works manages the batteries of the mobile robots. The proposed framework combines human-only, robot-only, and collaborative execution modes with mobile robot transport, relocation, and battery management, and is validated on a deployed industrial MMS; the comparison concerns the scope of the problem addressed, since differences in instances, objectives, and hardware prevent a direct comparison of solution quality or computation times. One observation is nonetheless consistent across the literature: Ferreira, Figueira, and Amorim [7] complemented their MILP formulation of the MMTSP with CP and GA solution methods, and most FJSSP works in Table 3.1 rely on metaheuristics. This reflects the same limitation observed here for the Task Scheduling module, namely that exact solution of the integrated problem becomes costly as instances grow, and supports the use of such methods for larger deployments (Section 7.3).

Table 5.18: Comparison of the proposed coordination framework with the closest related works reviewed in Section 3.1. Execution modes: H, human; R, robot; C, human-robot collaboration. "–": not addressed or not reported.

Study	Modes	Mobile transport	Battery mgmt.	Method	Validation	Reactive re-scheduling
[109]	–	Yes	–	PSO and simulated annealing	Benchmark	–
[119]	–	Yes	–	GA (energy objective)	Benchmark	–
[7]	H, R, C	–	–	MILP; CP; GA	Benchmark	–
[126]	H, R	–	–	GA	Gearbox assembly cell	–
[130]	H, R	–	–	AOG; A*	Assembly of metallic profiles	–
[125]	H, R	–	–	AOG; AO*	Assembly of metallic profiles	Yes
[131]	H, R	–	–	Time PT net; B&B	Assembly station	–
[20]	H, R	–	–	AOG; dynamic programming	Simulation	Yes
[25]	H, R	–	–	BT; MILP	Metallic structure assembly	–
This work	H, R, C	Yes	Yes	Three sequential MILP modules (Branch-and-Cut)	Deployed industrial MMS (alternator disassembly)	Yes

The computation times reported in Table 5.14 reveal an important asymmetry in the computational demands of the three modules. The Battery Swap and Mobile Robot Relocation modules resolved in under 0.5 s and 0.01 s respectively, reflecting their comparatively narrow solution spaces. The Task Scheduling module is the primary computational bottleneck of the pipeline, requiring 34.74 s in total for the first cycle. This is consistent with the formulation analysis in Section 5.3.2. Unlike the other two modules, task scheduling jointly optimizes allocation, sequencing, and timing variables across multiple agents and workstations, producing a solution space that grows combinatorially with problem size. While this remains tractable in the small-scale experimental use case considered here, it is the Task Scheduling module whose solve time is most likely to exceed the available scheduling window as the number of jobs, agents, and workstations increases in larger industrial deployments.

The source of this growth can be traced to the structure of the formulation. The allocation variables $Y_{a,j,o,m}$ grow linearly with the number of agent–job–operation–workstation combinations, but the sequencing variables $Z_{a,j,o,a',j',o',m}$ and $V_{a,j,o,m,j',o',m'}$ are defined over *pairs* of such combinations, so their number, and the number of no-overlap constraints (5.11c)–(5.11f) and priority constraints (5.13d)–(5.13e) built on them, grows approximately with the square of that number. The Battery Swap module has a similar pairwise structure ($U_{e,c,s_l,s_m,c',s'_l,s'_m}$), but over the much smaller sets of mobile robots and ABC slots, which explains its short solve times. In addition, all disjunctive constraints are written with the big-M technique (Section 2.2); a large constant M

weakens the linear relaxation used by Branch-and-Cut and therefore increases the number of subproblems explored. Setting M to the smallest valid value for each constraint, for example from the sum of processing and transport times over the scheduling horizon, is a direct way to tighten the formulation. Finally, the lexicographic strategy has a measurable cost: the flowtime stage accounted for 12.56 s of the 34.74 s of the Task Scheduling module in the first cycle (Table 5.14) and 3.17 s of 32.32 s in the second (Table 5.16). Where response time is critical, this stage can be time-limited or omitted, accepting a makespan-optimal schedule with a possibly larger flowtime; likewise, a time limit on the makespan stage would return the best schedule found so far, at the cost of the optimality guarantee.

The reactive design of the coordination pipeline partially mitigates the identified hardware constraints. By triggering a fresh scheduling cycle after each operation completion, the framework incorporates the updated system state, including current robot positions, battery levels, and workstation availability, and exploits timing windows in the physical execution that would be lost in a purely offline approach. The case study demonstrated this concretely. When operation 0 completed while MPC 3 was still in transit to the ABC, the new scheduling cycle was able to redirect MPC 3 to a different workstation upon leaving the ABC, avoiding a conflict with MPC 2 and saving 3 seconds in relocation makespan.

Nevertheless, several limitations inherent to the current implementation must be acknowledged. Regarding system responsiveness, the Prolynk platform does not provide real-time robot localization, and MPCs cannot be rerouted mid-transit, since rerouting is only possible when an MPC is indexed at a workstation or at the ABC. This physical constraint limits the granularity of the reactive coordination, as scheduling corrections can only take effect at discrete indexing points rather than continuously during execution. Regarding optimality, the sequential decomposition means that battery swap planning and task allocation are not jointly optimized. Because the battery swap assignment minimizes charging time at the nearest ABC, it may place a robot far from the workstation where it is subsequently needed, increasing transport times and potentially inflating makespan. This is an accepted consequence of the decomposition strategy, and it means that the framework does not guarantee globally optimal solutions. Addressing these limitations, particularly real-time localization, mid-transit rerouting capability, and the scalability of the Task Scheduling module, constitutes the primary direction for future development of the framework.

Chapter 6

Operator Interfaces for Human-Robot Collaboration in Matrix Manufacturing

Effective HRI in collaborative manufacturing depends on the operator's ability to understand robot behavior, anticipate upcoming actions, and coordinate task execution within a shared workspace. When this information is poorly conveyed, coordination becomes less efficient, trust may decrease, and both workload and interaction effort may increase [162], [163]. Interface design therefore plays a central role in collaborative robotics, not only as a means of presenting task instructions, but also as a mechanism for supporting operator awareness, timing, and coordination with the robot. This challenge is particularly critical in flexible production paradigms such as the MMS, where operators must respond to changing task conditions and coordinate with robotic agents under varying production states [5], [6], [164], and where Industry 5.0 places the human at the centre of the production process [3], [165].

This chapter presents the design, implementation, and comparative evaluation of two complete operator-interface configurations for HRC in a shared industrial workspace. The first is a conventional touch screen HMI, representative of those widely deployed in industrial collaborative workstations, where task instructions are presented through video and text on a fixed screen adjacent to the workspace. The second is an AR interface implemented on the HL2 HMD, which replaces fixed-screen delivery with holographic overlays anchored directly within the physical workspace, providing in-situ task guidance and spatial visualization of the robot's intended trajectory. Both interfaces follow the same task sequence, instructional content, and confirmation flow. They differ in how they communicate task guidance, robot actions, and workspace-related spatial information, but also in the input and feedback mechanisms inherent to each device, and in robot-intent cues that are available only in AR. The study therefore evaluates interface configurations as they would be deployed at a workstation, rather than visualization modality in isolation; the differences between the two configurations are documented explicitly in Section 6.5. The work complements the robotic coordination framework for human-robot teams in MMS [21], which ensured coordination among robots and equipment, by incorporating the human operator into the loop and investigating how the interface shapes the operator's understanding of task state and robot intent.

As discussed in Section 3.2, several studies have shown that AR can support operators in HRC, but they leave three questions open, which define the novelty of the work presented in this chapter. First, most AR-based HRC studies compare AR against a manual, paper-based, or no-interface baseline, or compare interaction techniques within AR; few compare AR against a functionally complete industrial HMI that already provides instructions, system status, and production control, which is the realistic alternative on a shop floor. Second, HRI quality is usually assessed through general usability and workload instruments, and is rarely linked to a theoretical account of what the operator needs to know about the robot. Here, the evaluation is framed by situation awareness, robot-intent legibility, and attention theory (Section 6.1), which predicts not only where AR should help but also where it should cost. Third, within-subject interface comparisons with a repeated task are affected by practice, yet interface and practice effects are seldom separated. The present study uses a crossover analysis that estimates both, and resolves execution time by human, robot, and collaborative task type.

Accordingly, the chapter addresses three chapter-level questions (Q6.1–Q6.3), which refine the thesis research questions of Section 1.3:

- **Q6.1:** How does the interface configuration affect the operator's awareness of robot actions and the perceived quality of working with the robot?
- **Q6.2:** Which interaction costs, in terms of workload, eye fatigue, and usability, accompany these effects?
- **Q6.3:** How do execution performance and errors differ between configurations once practice effects are accounted for?

Q6.1 and Q6.3 address thesis question RQ2 (to what extent the AR interface improves user experience and performance), whereas Q6.2, together with the interpretation in Section 6.8, addresses RQ3 (which factors drive these improvements or deteriorations).

The contributions of this chapter are threefold. First, it provides a counterbalanced within-subject comparison of two functionally matched but complete interface configurations in a scheduler-driven collaborative disassembly workstation, with an explicit account of their non-equivalences. Second, it provides empirical evidence, obtained with crossover mixed models and multiplicity control, that the AR configuration improves awareness of the robot's next action and the perceived quality of working with the robot while increasing eye fatigue and reducing usability, and that practice effects on execution time are of the same order as interface effects. Third, it offers a situation-awareness-based interpretation of this trade-off, supported by task-type-level evidence that interaction overhead rather than task comprehension drives time and error differences, and derives design implications for hybrid interfaces in industrial HRC.

The remainder of the chapter is structured as follows. Section 6.1 introduces the theoretical framing and Section 6.2 positions the study with respect to prior work. Sections 6.3 to 6.6 describe the two interface configurations, their differences, and their extension to other products. Section 6.7 details the experimental design, measures, statistical analysis, and results, which are

interpreted in Section 6.8. Section 6.9 discusses limitations and future research, and Section 6.10 concludes the chapter. A video of both interfaces in use during the collaborative disassembly is available at `https://youtu.be/IztmLEqxkxg`.

6.1 Theoretical Framing: Situation Awareness, Robot Intent, and Attention

The concept of Situation Awareness (SA) is commonly defined as the perception of elements in the environment, the comprehension of their meaning, and the projection of their status in the near future [166]. In HRC, these three levels map naturally onto what the operator must know about the robot: its current state and position (Level 1), what this state implies for the operator's own task, including which agent owns which step (Level 2), and where and when the robot will move next (Level 3). Losing SA at any of these levels forces the operator either to monitor the robot continuously, diverting attention from the task, or to act without adequate anticipation, which degrades coordination and perceived safety.

Level 3 SA depends on how legible the robot's intent is. Robot motion is legible when an observer can infer the robot's goal from the motion itself [167]; industrial manipulators executing planned trajectories are often not legible in this sense, so intent must be externalized by the interface. AR can render planned trajectories and goals directly in the workspace: for a robot arm, a mixed-reality HMD visualization of the planned motion allowed users to judge where the robot would move more quickly and accurately than a 2D display visualization or no visualization [168], and for aerial robots, AR intent cues improved task efficiency over a baseline without them [169]. More generally, Suzuki, Karim, Xia, *et al.* [170] surveyed 460 papers on AR and robotics and classified them by, among other aspects, the purpose of the AR interface and the type of information it displays; robot-intent visualizations such as trajectory previews are one case within this broader body of work. A scoping review of robot motion-intent communication further characterizes intent cues by the type of intent, the information conveyed, and the location at which it is presented [171]; the last dimension is central to this chapter, since the AR and touch screen configurations convey equivalent task information at different locations. Anticipating the partner's actions is, in turn, a prerequisite for fluent collaboration, i.e., the well-synchronized meshing of the partners' actions [172].

The spatial location of information matters as well. When instructions are presented on a fixed screen separated from the task, the operator must alternate gaze between screen and workspace and mentally integrate the two sources, which increases cognitive cost (the split-attention effect) [173]; displays that place information close to the elements it refers to reduce this cost when the information must be integrated [174]. In-situ AR guidance is designed to exploit this principle. At the same time, the theory also predicts costs. Optical see-through HMDs present content at a fixed focal distance, and the resulting vergence-accommodation conflict is a known source of visual fatigue [175]; dense holograms may occlude the physical parts that must be manipulated; and mid-air input without tactile confirmation shifts effort from task execution to interface operation.

This framing yields three expectations that guided the selection of dependent measures:

- **E1:** In-situ robot-intent cues should improve robot-related SA indicators, particularly awareness of the robot's next action and not losing track of the robot, and should strengthen the perceived sense of collaboration.

- **E2:** HMD viewing and mid-air input should increase eye fatigue and interaction effort, thereby reducing usability.

- **E3:** Because both interfaces share the same confirmation flow, differences in execution time and errors should mainly reflect input and feedback overhead rather than differences in task comprehension.

Table 6.1 maps each construct to the interface features of both conditions and to the corresponding measures.

Table 6.1: Mapping between theoretical constructs, interface features, and measures. WWR: Working with the Robot block; ARV: AR Visualization block; GEN: General Questions block (Appendix A.4).

Construct	**AR configuration**	**Touch screen configuration**	**Measures**
Level 1 SA: robot state	Robot status on the floating slate; holograms anchored to the robot	Textual robot goals and status on the screen	WWR item 2 (lost track); ARV item 1; interviews
Level 2 SA: task ownership	Color-coded components (green: operator; red: robot)	Textual indication of the agent in charge	WWR items 2 and 6; interviews
Level 3 SA: projection of robot motion	In-situ preview of the planned trajectory before and during motion	Pre-recorded instructional video before execution	WWR item 1 (awareness of next action); ARV items 2–3
Attention allocation	Instructions anchored to the physical parts	Instructions on a fixed screen outside the task area	WWR item 4; GEN item 4; execution time
Perceived collaboration and fluency	*Outcome of the SA-related features above; no dedicated feature*		WWR items 3, 5, and 6
Interaction cost	Mid-air pinch and virtual buttons; no haptic feedback; HMD viewing	Touch input with tactile confirmation; conventional display	SUS; NASA-TLX; eye-fatigue item; errors

6.2 Positioning with Respect to Prior Work

6.2.1 Interface Configurations in Human-Robot Collaboration

Beyond the AR-based HRC studies reviewed in Section 3.2, a recent survey of AR for HRC reviews the AR technologies, application areas, and evaluation methods used in the field, with recommendations for their implementation [27]. Multimodal HRI has also been surveyed in the context of human-centric smart manufacturing [176], which is especially relevant in Industry 5.0, where the wellbeing of the worker is placed at the center of the production process [3]. Reviews of HRC in the context of Industry 5.0 identify AR, virtual reality, and digital twins among the technologies that enable human-centric collaboration [177], and industrial-scale implementations are emerging: Yang, Amtsberg, Kaiser, *et al.* [178] combined head-mounted AR for multiple workers with dynamic human task allocation and safety-zone generation for high-payload robots in timber prefabrication, and evaluated the system through simulation, a user study, and an industrial demonstrator.

Across these works, AR is typically evaluated either as an enabling layer within a new system or by comparing intent cues with their absence; comparatively few studies ask how an AR configuration performs against the conventional screen-based HMI already present at industrial workstations, which supplies the same task information through a different channel. Taken together with the studies summarized in Table 3.3, the evidence base shows three limitations. First, the reference condition is often a manual or no-interface baseline [25], [55], [130], or the comparison is made between interaction techniques within AR [19], [138]. Where non-HMD displays are included (a projector [55], screen-based feedback [54], or a tablet [26]), they typically deliver instructions or feedback only, rather than the full set of monitoring and production-control functions that an industrial HMI provides; the resulting advantage of AR may therefore partly reflect a weak baseline. Second, HRI quality is mostly assessed with general usability and workload instruments, while measures tied to what the operator knows about the robot, such as awareness of its next action or of task ownership, are rare; Lunding, Lystbæk, Feuchtner, *et al.* [26] are a notable exception and provide the item basis for the questionnaire used in this chapter. Third, within-subject designs in which the same task is repeated seldom separate interface effects from practice effects, although learning is substantial in short assembly tasks. In addition, the reviewed studies focus almost exclusively on assembly, whereas disassembly with explicit human, robot, and collaborative steps has received little attention from an interface perspective.

This chapter addresses these limitations by comparing an AR interface with a touch screen HMI that replicates the functional structure of an industrial operator interface, within the same collaborative disassembly task and with the same task sequence, instructional content, and confirmation flow. Because the two configurations still differ in visualization content, display, input, and feedback, observed differences are attributed to the interface configuration as a whole; the specific features involved are identified explicitly (Section 6.5) and linked to the results through item-level and interview evidence. Performance is analyzed with a crossover model that separates interface

from practice effects, and resolved by human, robot, and collaborative task type.

6.2.2 Collaborative Disassembly and Task Allocation

Disassembly is central to the end-of-life treatment of products: without an appropriate disassembly plan, products such as waste electrical and electronic equipment are disposed of unsafely [179]. The literature distinguishes complete disassembly, in which all components are separated, from partial or selective disassembly, in which only target components (e.g., valuable, hazardous, or reusable parts) are removed. In this context, Disassembly Sequence Planning (DSP) determines the disassembly level and the order of operations under precedence, geometric-feasibility, and stability constraints, and is typically formulated as a combinatorial optimization problem. Recent formulations extend the objective beyond disassembly time and cost: multi-layered methods support de-manufacturing decisions by extracting valuable materials while keeping the separation of toxic elements within threshold limit values [180], stability-graph cut-set methods generate the optimal disassembly sequence together with the disposal of the parts [181], and mechanical DSP has been formulated to maximize the recyclability of end-of-life products [182]. AR has also been applied to disassembly guidance: Eswaran and Raju Bahubalendruni [183] proposed an AR/VR-assisted framework that simulates the disassembly in a virtual platform, renders the corresponding virtual scenes, and generates the model targets needed to register them; these are then used in the Unity AR tool to create AR guidance for the disassembly operations. The work is motivated by the time and trained AR specialists that the manual creation of AR instructions requires. In human-robot collaborative disassembly, an additional decision concerns the allocation of each operation to the human, to the robot, or to both agents, which in the present system is handled by the scheduler of the workstation [21].

This chapter does not address DSP. The alternator used in the experiment (Section 6.7.1) is completely disassembled following a fixed, feasible sequence, and each operation is assigned in advance to the operator (H), the robot (R), or both (C). This separation is deliberate: the interfaces are consumers of a sequence and an allocation, not generators of them. Because the steps, their order, and the admissible agents are read from a database at runtime (Section 6.6), sequences produced by a DSP method, whether for complete, partial, or selective disassembly, could be supplied to either interface without changing its logic.

6.3 Non-Immersive Interface

The touch screen HMI was developed to reflect the type of operator interface commonly deployed in industrial collaborative workstations, where a screen-based system serves as the primary channel through which the operator receives task instructions, monitors system state, and exchanges information with the production system and the robot. The interface was designed to address the actual operational demands of the collaborative workstation and to serve as a methodologically credible baseline for the comparison, since it replicates the same task logic, instructional content,

and workflow progression as the AR interface, differing in how and where that information is presented and in the input and feedback mechanisms of the device.

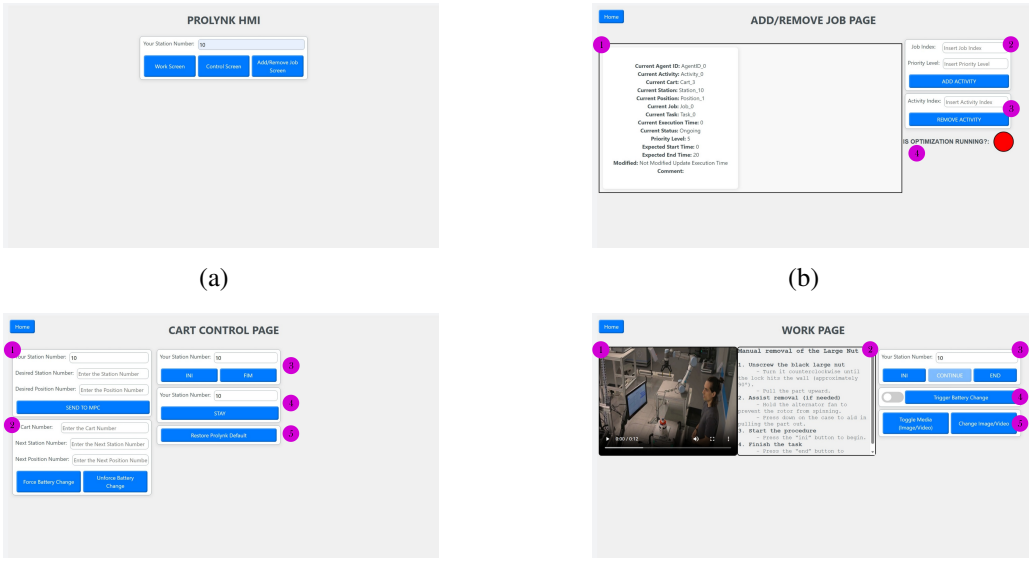
A central design decision was the separation of the interface into four functionally distinct pages, each targeting a different level of operator activity. This structure responds to a concrete operational need. In a collaborative workstation embedded within a scheduled MMS, the operator must simultaneously manage production-level interactions (adding or removing jobs, monitoring the scheduling algorithm), system-level interventions (routing MPCs, issuing hold commands, managing battery replacements), and execution-level actions (following task instructions, coordinating with the robot, signaling step completion). Collapsing these into a single screen would create contextual overload, forcing the operator to parse irrelevant information during time-sensitive task execution. The four-page structure prevents navigational, supervisory, configurational, and execution-level interactions from competing for attention simultaneously, and helps the operator maintain clarity about which level of the system they are currently acting on. The interface was developed in Node-RED, integrating *node-red-contrib-flowake-ros-nodes*[14] for ROS connectivity, *node-red-contrib-opcua*[15] for OPC-UA communication, and *node-red-contrib-uibuilder*[16] for hosting and rendering, enabling seamless communication with the Prolynk system across all pages. The four pages are shown in Fig. 6.1.

The Main page (Fig. 6.1a) serves as the entry point, where the operator enters the station number to ensure all subsequent pages display the correct station-specific information, and provides navigation buttons to the remaining sections. The Jobs page (Fig. 6.1b) connects the operator to the production scheduling system, displaying ongoing activities with relevant process data, such as current station, elapsed task time, assigned operator, active MPC, and contextual information (Fig. 6.1b.1). It also provides panels for adding or removing tasks (Fig. 6.1b.2, 6.1b.3) and a status indicator showing whether the scheduling algorithm is active, informing the operator of when job actions will be processed (Fig. 6.1b.4). Together with the Control page, this gives the operator visibility into and control over the broader production system, establishing the touch screen as more than an instruction display, which is what makes it a credible industrial HMI baseline rather than a passive visual support tool. The Control page (Fig. 6.1c) supports manual intervention independently of the scheduling algorithm, covering MPC routing (Fig. 6.1c.1), battery replacement at the ABC (Fig. 6.1c.2), manual procedure execution without triggering the optimizer (Fig. 6.1c.3), station hold commands (Fig. 6.1c.4), and PLC flag reset to ensure clean disconnection (Fig. 6.1c.5); all commands are executed only when the scheduling algorithm is idle, preserving system integrity under automated operation.

The Work page (Fig. 6.1d) is the layer most directly relevant to the comparison. It concentrates everything the operator needs during task execution, instructional video and images (Fig. 6.1d.1), textual guidance describing robot goals and operator actions (Fig. 6.1d.2), task progression controls for signaling step start and completion (Fig. 6.1d.3), a battery-change trigger for the assigned

[14] https://flows.nodered.org/node/node-red-contrib-flowake-ros-nodes
[15] https://flows.nodered.org/node/node-red-contrib-opcua
[16] https://flows.nodered.org/node/node-red-contrib-uibuilder

MPC (Fig. 6.1d.4), and a media toggle between image and video formats (Fig. 6.1d.5). This page is the functional equivalent of the AR Work Page slate. Both deliver the same instructional content and task control logic; they differ in the channel through which that content reaches the operator (screen-based and spatially separated from the workspace in the touch screen condition, spatially anchored within the workspace in the AR condition), in the robot-intent cues available, and in how the operator provides input (Section 6.5).



(a) (b)

(c) (d)

Figure 6.1: Touch screen HMI illustrating the main interaction pages: (a) main dashboard, (b) job management page displaying task allocation and scheduling information, (c) control interface for system operation and command execution, and (d) work page supporting operator interaction during task execution.

6.4 Immersive Interface

6.4.1 Design Rationale

The AR interface was conceived to address specific limitations of screen-based instruction delivery in collaborative human-robot tasks. When an operator must consult a fixed screen for task guidance while simultaneously monitoring a robot moving in a shared workspace, attention is divided between two spatially separated information sources. This separation increases the cognitive cost of maintaining SA and can reduce the operator's ability to anticipate robot actions, interpret task ownership, and coordinate timing within the shared workspace (Section 6.1). The AR interface was therefore designed around three interaction needs identified as central to HRC quality: delivering task instructions within the physical workspace so that the operator's attention remains anchored to the task scene; making the robot's intended trajectory and upcoming actions spatially legible before and during movement; and preserving the same instructional logic and task control

structure as the touch screen baseline, so that differences between conditions can be attributed to the interface configuration rather than to task content. This design rationale reflects the design intent that the reviewed literature identifies as a key advantage of AR in HRC contexts, namely improved spatial awareness, clearer robot intent communication, and stronger perceived coordination, and it directly anticipates the dimensions evaluated in the user study: workload, usability, robot action awareness, perceived collaboration, and execution performance. The trade-offs described at the end of this section are acknowledged as inherent consequences of these choices rather than implementation deficiencies.

6.4.2 Technical Implementation

The interface was developed in Unity (v2022.3.62f) using the MRTK (v2.8.3). Stable spatial alignment between holographic content, the robot, and the physical assembly components is a prerequisite for the interface to function as intended. Misalignment of AR content relative to its physical counterpart can significantly impact user performance, depending on task complexity and nature [86]. To address this, a QR code is placed in the work area at a known offset relative to the robot, establishing a shared coordinate system between the AR application and the physical workspace. The HL2 leverages its built-in QR code reader together with world-understanding and SLAM capabilities to maintain accurate hologram alignment under natural operator movement. This marker-based tracking approach was selected for its practical advantages in an industrial setting, rapid recalibration, low computational overhead, and native support on the HL2 [17], all of which contribute to the plausibility of the solution as an industrial deployment. Communication between the HL2 and the ROS server is handled via *ROS-TCP Connector*, enabling real-time data exchange through the ROS bridge so that holographic content reflects the live state of the robot and the production system at all times.

To reduce the learning cost associated with a novel interaction modality and limit the influence of unfamiliarity with AR interaction, the interface includes an initial tutorial that emulates the actual disassembly procedure. This tutorial systematically demonstrates all system functionalities and clarifies how tasks handled exclusively by the operator, exclusively by the robot, or collaboratively should be performed before the experimental session begins. An equivalent familiarization was provided for the touch screen (Section 6.7.2.2).

6.4.3 Interface Structure and Human-Robot Interaction Relevant Features

The AR interface is organized around a floating slate that mirrors the functional structure of the touch screen, providing access to job management, system control, and task guidance through equivalent pages, ensuring that the instructional logic and task control flow remain comparable across both conditions. The figures (Fig. 6.2a, 6.2b, 6.2c, 6.2d) use the same numbering as Fig. 6.1 to facilitate direct comparison.

Interaction with the floating slate is performed through the HL2's enabled input modalities: near-field interaction, where the operator reaches out and touches virtual buttons with their index

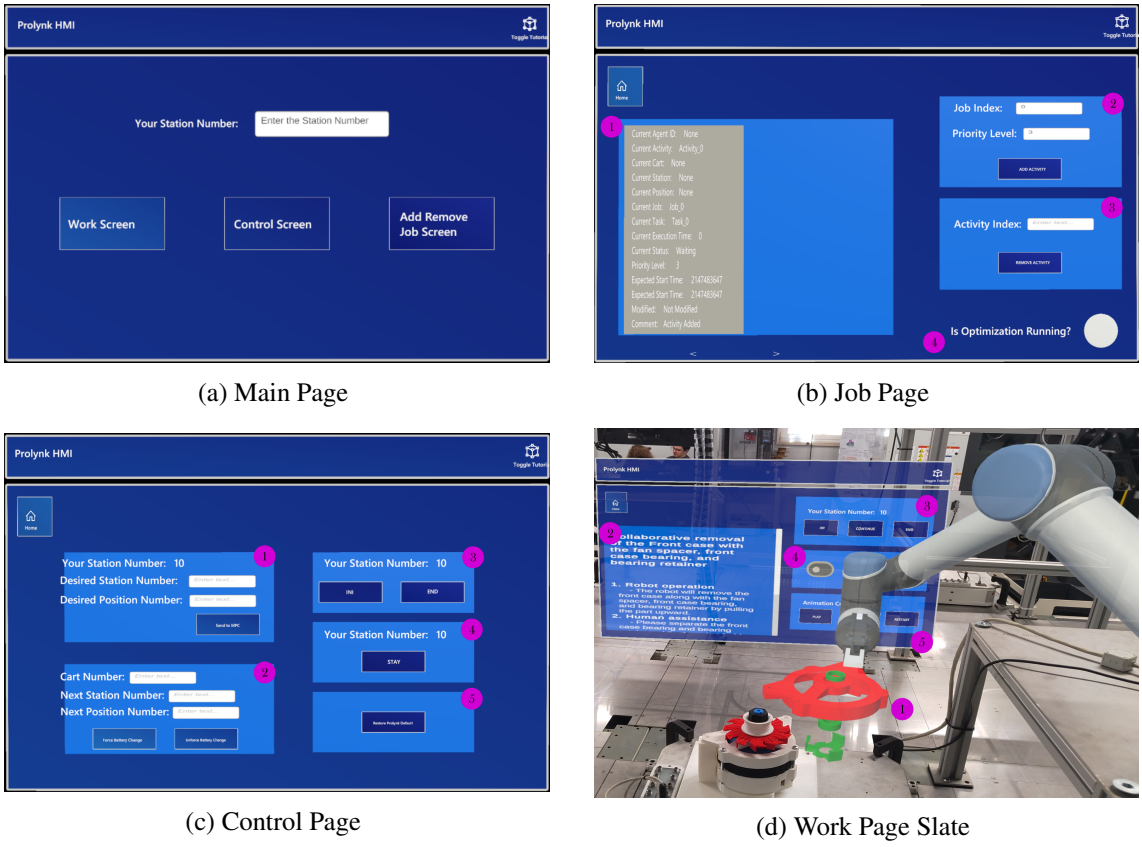
finger, and far-field interaction via hand-ray pointing, where a ray projected from the hand is used to select elements at a distance. The MRTK provides built-in support for both modalities, as well as for eye-gaze-based focus and voice commands, though the interface relies primarily on hand-based input for consistency and reliability in the industrial context. These interaction methods differ fundamentally from touch screen interaction in ways that have direct usability implications. Near-field virtual button presses lack the haptic feedback that confirms a successful press on a physical screen, which can introduce uncertainty about whether an input has been registered. Far-field hand-ray selection requires sustained arm extension and fine motor stability, which can increase physical demand over repeated interactions. The virtual keyboard required for text input represents a particular interaction cost, as mid-air typing is considerably slower and less reliable than physical keyboard or touchscreen input. These are accepted trade-offs of the AR configuration rather than design oversights, and their effects on workload and usability are expected to surface in the experimental evaluation (expectation E2).

The key differentiator from the touch screen lies in the Work Page. Rather than presenting static images or videos on a fixed screen, the AR interface overlays animated step-by-step holograms directly onto the physical workspace (Fig. 6.2d.1), aligning virtual elements with their real counterparts so that the operator receives task guidance while looking at the actual components and the robot. This in-situ delivery is designed to reduce the attentional cost of switching between screen and workspace and to make task actions spatially legible without requiring the operator to mentally map screen content onto the physical scene.

The same spatial framework provides visualization of the intended trajectory of the UR5 robotic manipulator before and during its movement, offering information about robot intent that has no direct equivalent in the touch screen interface. In the touch screen condition, the operator receives information about robot goals through textual instructions, and can observe the planned robot trajectory through the instructional video prior to task execution. The trajectory preview is therefore the primary mechanism through which the AR interface is expected to improve robot action awareness and support coordination timing, by allowing the operator to perceive the robot's intended path directly within the physical scene. Color coding further supports task ownership clarity: components assigned to the operator are highlighted in green, while parts handled by the robot appear in red, making the division of labor between agents immediately readable from the operator's perspective.

However, these spatial advantages introduce a set of design trade-offs. The holographic overlays applied to physical components are rendered at a fixed opacity level, which may reduce visibility of the underlying parts during manipulation, particularly when the hologram color closely matches the physical object. The limited field of view of the HL2 may hide interface elements and holographic content positioned outside the central viewing area, requiring the operator to move their head and potentially disrupting task flow. Prolonged use of the device introduces ergonomic considerations, since the headset weight and the sustained arm postures required for mid-air interaction can contribute to physical fatigue. Finally, the floating slate and the Work Page holographic content compete visually for operator attention within the same field of view,

which requires careful spatial placement to maintain a clear visual hierarchy. These factors reflect fundamental characteristics of current HMD technology that any realistic deployment of AR in collaborative manufacturing must account for, and they directly impact the workload, usability, and eye-fatigue dimensions of the experimental evaluation presented in Section 6.7.



(a) Main Page

(b) Job Page

(c) Control Page

(d) Work Page Slate

Figure 6.2: AR HMI illustrating the main interaction components: (a) main page providing an overview of system status and available functionalities, (b) job page displaying task allocation and scheduling information within the augmented environment, (c) control page enabling system operation and command execution through virtual elements, and (d) work page slate presenting context-aware instructions and guidance to support the operator during task execution.

6.5 Differences Between the Interface Configurations

The two interfaces were designed to be functionally equivalent at the level of the task, but, as in any comparison between real devices, they are not equivalent at the level of the interaction. Table 6.2 lists what the configurations share and where they differ. The shared elements are the disassembly sequence, the instructional content of each step, the agent assignment, the robot program, the scheduler, and the confirmation flow, in which the operator signals the start (INI), the completion (END), and, where required, the continuation (CONTINUE) of each operation. The differences fall into three groups: (i) visualization content, since the trajectory preview and the color-coded task ownership exist only in AR; (ii) display, i.e., a fixed screen outside the task area versus holograms

anchored in the workspace and viewed through an HMD with a limited field of view; and (iii) input and feedback, i.e., touch input with tactile confirmation versus mid-air pinch gestures and virtual buttons without haptic feedback.

The comparison therefore estimates the effect of the complete AR configuration relative to the complete touch screen configuration. It does not isolate the contribution of any single feature; for example, color-coded task ownership could in principle be added to a screen-based interface. Links between specific features and specific outcomes are thus drawn only where item-level and interview evidence point to them, and are interpreted as hypotheses for feature-level studies (Section 6.9).

Table 6.2: Shared elements and differences between the two interface configurations.

Aspect	**Touch screen configuration**	**AR configuration**
Shared		
Task, sequence, agents	Same alternator, same nine operations, same fixed H/R/C allocation and robot program	
Instructional content	Same step descriptions and same robot goals for each operation	
Confirmation flow	Same INI/END/CONTINUE sequence triggering the scheduler and robot motion	
System functions	Same four pages (Main, Jobs, Control, Work) connected to the same ROS/OPC-UA back-end	
Different		
Instruction medium	Pre-recorded video, images, and text	Animated 3D holograms of robot and parts, and text on a floating slate
Location of information	Fixed screen outside the task area	Anchored to the physical parts and robot in the workspace
Robot intent	Textual goals; trajectory visible only in the pre-recorded video	In-situ preview of the planned trajectory before and during motion
Task ownership	Textual indication of the agent in charge	Color-coded components (green: operator; red: robot)
Input	Touch input on a physical screen	Mid-air pinch gestures, close-range virtual buttons, virtual keyboard
Input feedback	Tactile contact with the screen	Visual only; no haptic confirmation
Viewing	Conventional display; all information visible at once	Optical see-through HMD; limited field of view requiring head movement
Familiarization	Guided hands-on walkthrough with a researcher	In-application tutorial emulating the procedure

6.6 Content Authoring and Extension to Other Products

Both interfaces draw on the same product description stored in the system database, which specifies the disassembly steps, their sequence, the agents permitted to perform each step, and the file path of the associated animation (AR interface) or video (touch screen interface). The scheduler, the communication layer (ROS and OPC-UA), the page structure, and the confirmation logic are product-independent. Extending the system to a new product or product variant therefore requires (i) registering its steps, order, and admissible agents in the database, and (ii) authoring the step content: textual instructions and instructional videos for the touch screen, and 3D models and animations for the AR interface. In the current implementation, the AR animations are authored in Unity and built into the application, so that adding a product requires rebuilding and redeploying the AR application, whereas touch screen content can be added without modifying the software.

Content authoring is therefore the main factor limiting the scalability of the AR configuration to a large number of product configurations, and its cost grows with the number of products and variants along with the complexity of the scheduling problem. Variants that share components can reuse models and animations, and the marker-based registration (Section 6.4.2) only requires the offset between marker and robot to be updated when the station layout changes. Loading animation assets at runtime instead of at build time, and generating them from CAD data and the disassembly sequence, as in frameworks that render the virtual scenes and model targets for AR disassembly guidance in a virtual platform [183], are natural next steps to reduce this cost; neither was evaluated in the present study.

6.7 Experimental Validations and Results

This section presents the comparative evaluation of the two interface configurations described above. It details the study design, participants, procedure, dependent measures, and statistical analysis, followed by the results. The evaluation is organized around the three research questions stated at the beginning of the chapter: the effect of the interface configuration on robot awareness and the perceived quality of working with the robot (Q6.1), the interaction costs that accompany it (Q6.2), and the differences in execution performance and errors once practice effects are accounted for (Q6.3).

6.7.1 Study Design

This study investigates how the interface configuration affects HRI quality in a collaborative disassembly task within an MMS. Rather than evaluating AR as a technology in isolation, the study compares the AR interface against the touch screen baseline to examine how each configuration shapes operator awareness of robot behavior and upcoming actions, perceived coordination with the robot, clarity of task instructions, cognitive workload, usability, and execution performance, both at the overall system level and with respect to specific interface features. A within-subject design was adopted, in which every participant performed the task with both interfaces in one of two

counterbalanced orders (a two-period, two-sequence crossover design), so that each participant served as their own control and differences in individual skill did not confound the comparison.

The experimental setup was deliberately scoped to a single collaborative workstation rather than the full MMS workflow. A single MPC is deployed and remains stationary, with participants performing all disassembly operations sequentially at the collaborative station. The arrival of new tasks is simulated, and the assignment of each operation to a specific agent was fixed in advance rather than determined dynamically by the scheduling algorithm. By removing variability introduced by multi-station routing, dynamic task allocation, and MPC circulation, the comparison between interfaces is not confounded by differences in system-level complexity unrelated to the interface itself. This approach trades ecological completeness for internal validity, ensuring that observed differences in operator awareness, workload, usability, and execution performance can be attributed to the interface configuration rather than to scheduling or workflow factors.

The experiment uses the same 3D-printed car alternator model shown in Fig. 5.2. The task consisted of the complete disassembly of the alternator following a fixed, predefined sequence; no DSP was performed (Section 6.2.2). The task was chosen to ensure representation of all three interaction modes relevant to collaborative robotics: steps performed exclusively by the human operator (H), steps executed solely by the UR5 robot (R), and steps requiring coordinated interaction between both agents (C). This coverage is methodologically important because each interaction mode places different demands on the operator in terms of coordination, robot awareness, and timing in the shared workspace. Human-exclusive steps test whether the interface supports task comprehension and workflow progression independently of the robot; robot-exclusive steps test whether the operator can monitor and interpret robot behavior with sufficient awareness to confirm completion and proceed; and collaborative steps test whether the interface supports coordination between the operator and the robot in a physically shared task. The agent assignment of each component was fixed in advance, even in cases where multiple agents were technically capable of performing the task, ensuring consistent task structure across participants and interfaces. Detailed information on the disassembly components, task sequence, and agent assignments is provided in Table 6.3.

Table 6.3: Components, disassembly sequence, and responsible agent types (Human - H, Robot - R, Collaborative - C) for the Car Alternator disassembly case study.

Components	**Disassembly Order**	**Disassembly Agent**
Large nut	0	H
Pulley	1	R
Fan	2	R
Case screws and nuts	3	H
Front case, fan spacer, front case bearing, and bearing retainer	4	C
Rotor	5	R
Stator	6	R
Electronics	7	H
Rear case and rear case bearing	8	C

As highlighted in Table 3.3, qualitative analyses remain more common than quantitative evaluations in experimental studies of AR-based HRC interfaces [9]. Among qualitative approaches, the SUS and NASA-TLX questionnaires are most frequently used to assess system usability and cognitive workload respectively, while custom questionnaires capture subjective aspects such as perceived collaboration, trust, and HRI quality. Quantitative metrics typically focus on task execution time and error rates. This study therefore evaluated both dimensions. Operational performance and accuracy were assessed through execution time and error rates, while cognitive workload, usability, and the perceived quality of human-robot interaction were measured using the NASA-TLX, SUS, custom questionnaires, and semi-structured interviews.

6.7.2 Study Protocol

The experiments were carried out at INESC TEC's Industry and Innovation Lab (iiLab)[17]. All procedures adhered to ethical research standards and data protection guidelines established by INESC TEC's ethics committee, which also formally approved the study with reference 1/25. These guidelines include: (i) respecting participants' autonomy, (ii) minimizing potential harm, and (iii) ensuring privacy and data protection. Participation in the study was entirely voluntary, and participants were free to withdraw at any time. No compensation was offered for their participation. Additionally, researchers were present throughout the sessions to ensure the smooth execution of the experiment and to provide assistance if needed.

6.7.2.1 Participants

The study involved 36 participants (mean age: 30.9 years), of whom 83% were male and 17% female. Participants represented a range of educational backgrounds, with the majority holding a Master's degree (53%), followed by PhD (22%) and Bachelor's degrees (19%), as shown in Fig. 6.3, Fig. 6.4, and Fig. 6.5.

Most participants had a technical background, primarily in engineering (81%) and computer science or information technology (17%). Regarding professional status, 56% were employed full-time, mainly as researchers or engineers, 42% were students, and 2% retirees, as shown in Fig. 6.6, Fig. 6.7, and Fig. 6.8. All participants were recruited in the same way, through an e-mail invitation. Five participants were invited in advance as domain experts (hereafter, experts): four hold a PhD and one is a PhD candidate in XR and/or interface design, and all rated their AR/VR experience as above average or excellent. The experts followed exactly the same protocol and interview as the other participants and are included in all quantitative analyses; their interview statements are reported separately only where they add a specialist perspective.

With respect to vision, 61% of participants reported using visual correction. Among those, 95% wore glasses and 5% contact lenses. Approximately 23% of participants who required visual correction chose not to use it during the experiment, as shown in Fig. 6.9 and Fig. 6.10. This

[17] https://www.inesctec.pt/pt/laboratorios/iilab-laboratorio-de-industria-e-inovacao

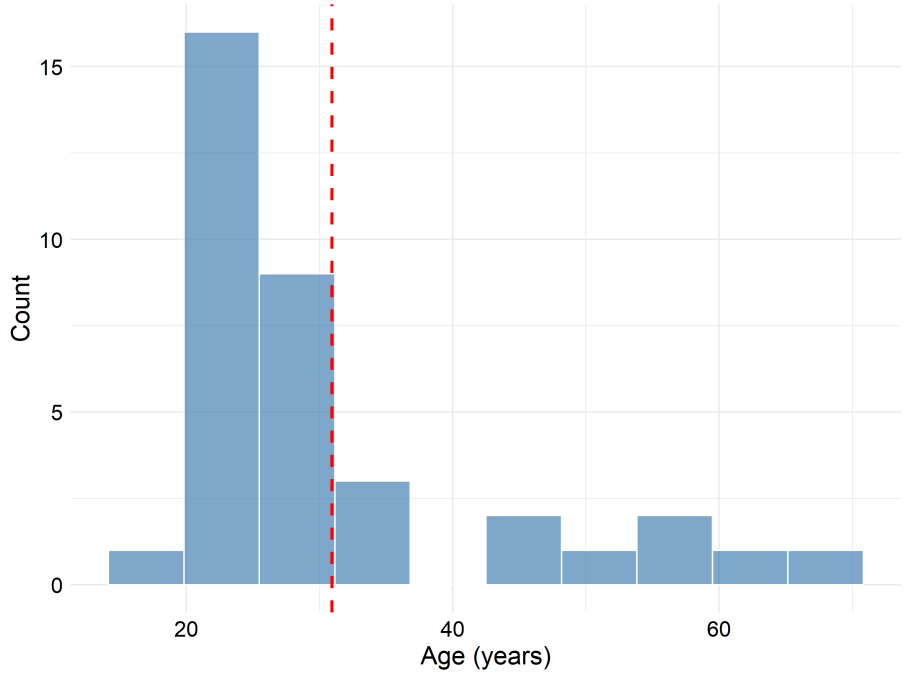


Figure 6.3: Distribution of participants' ages in the experimental study, providing an overview of the demographic range considered in the evaluation.

factor is of particular relevance to the eye-fatigue analysis, as prolonged HMD use may interact differently with participants depending on their visual correction needs.

Participants exhibited a broad range of prior experience with AR/VR systems, industrial assembly and disassembly procedures, and robotic systems, spanning all five experience levels (poor, below average, average, above average, and excellent). Experience with AR/VR systems was widely distributed, with equal proportions of participants reporting above-average (25.0%) and below-average (25.0%) experience, while 22.2% reported poor experience and 11.1% reported excellent familiarity. For industrial assembly and disassembly procedures, most participants reported average to above-average experience (52.8%), whereas 19.4% indicated poor experience and only 5.6% reported excellent proficiency. In contrast, experience working with robots was generally high: 77.8% of participants reported above-average or excellent experience, while only 11.1% reported average experience and fewer than 12% reported below-average or poor familiarity, as seen in Fig. 6.11, Fig. 6.12, and Fig. 6.13. The breadth of experience levels allowed interface effects to be explored across users with different backgrounds; these subgroup analyses are exploratory (Section 6.7.3.2).

No a priori power analysis was performed; the sample size was determined by the availability of participants during the experimental campaign, and is twice that of the most closely related study, which compared AR and tablet interfaces with 18 participants [26]. A sensitivity analysis for a two-sided paired t-test ($\alpha = 0.05$, $n = 36$) indicates 80% power for within-subject effects of $d_z \geq 0.48$, i.e., approximately medium effects according to Cohen's conventions [184]. The test for differential carryover, which compares participant totals between the two sequences (18 versus

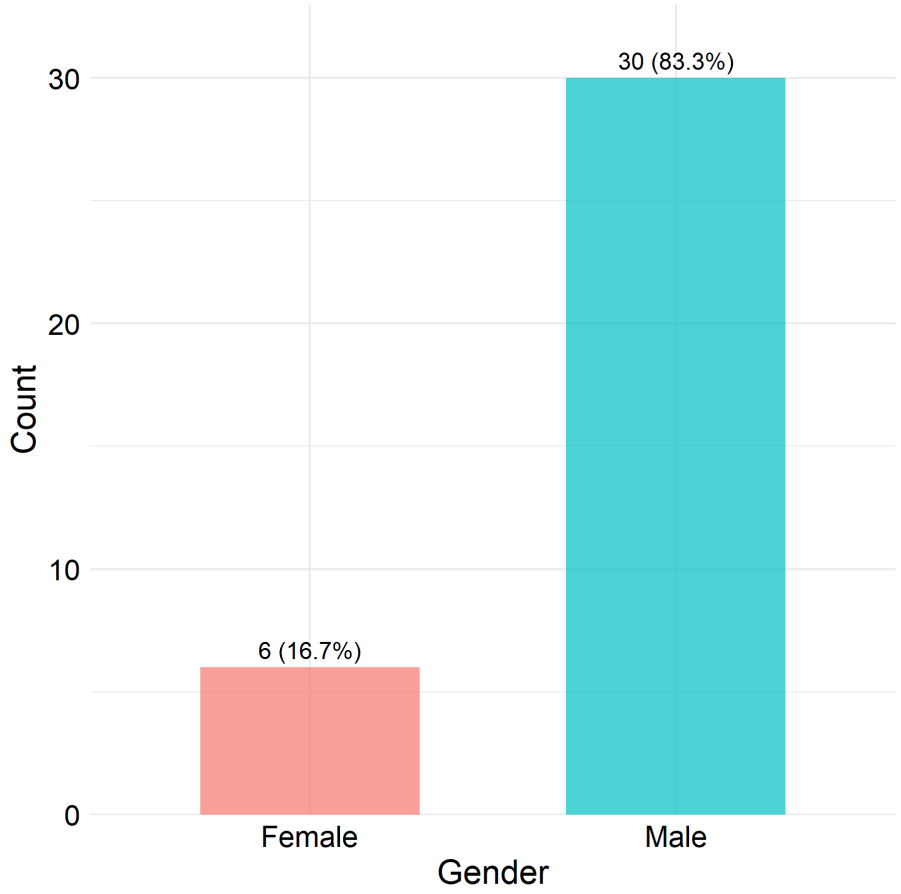


Figure 6.4: Gender distribution of participants involved in the experimental study, supporting the characterization of the sample population.

18 participants), has 80% power only for effects of $d \geq 0.96$, i.e., beyond Cohen's threshold for large effects. Within-subject effects smaller than $d_z = 0.48$, and carryover effects of moderate size, may therefore have gone undetected. The sample is also predominantly male, engineering-oriented, and experienced with robots, which limits generalization to industrial operators; this is discussed in Section 6.9.

6.7.2.2 Procedure

The experiment began with an introductory briefing outlining the research objectives and the disassembly task, after which participants signed a consent form and completed a short demographic questionnaire (Appendix A.1.1). Each participant was randomly assigned to either Group A or Group B using a counterbalanced design to mitigate first-order carryover effects, following a Latin Square arrangement based on Williams' design [185], [186]. With two conditions, this arrangement reduces to the two sequences touch screen→AR and AR→touch screen, each assigned to 18 participants. Group A began with the touch screen interface, while Group B started with the AR system. Before each session, participants were familiarized with the interface to be used. For

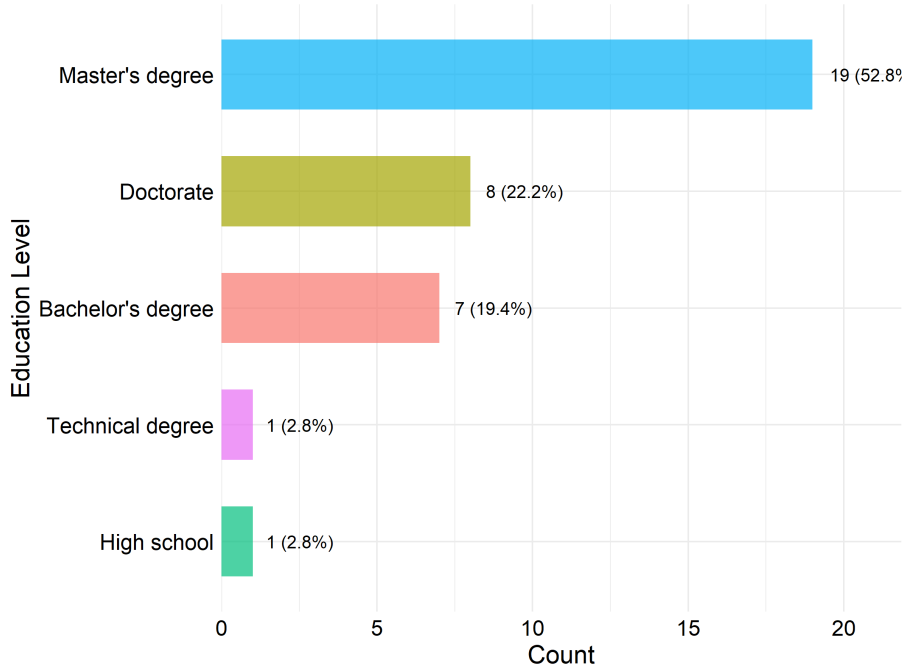


Figure 6.5: Educational background of the participants, indicating the level of academic qualification within the study sample.

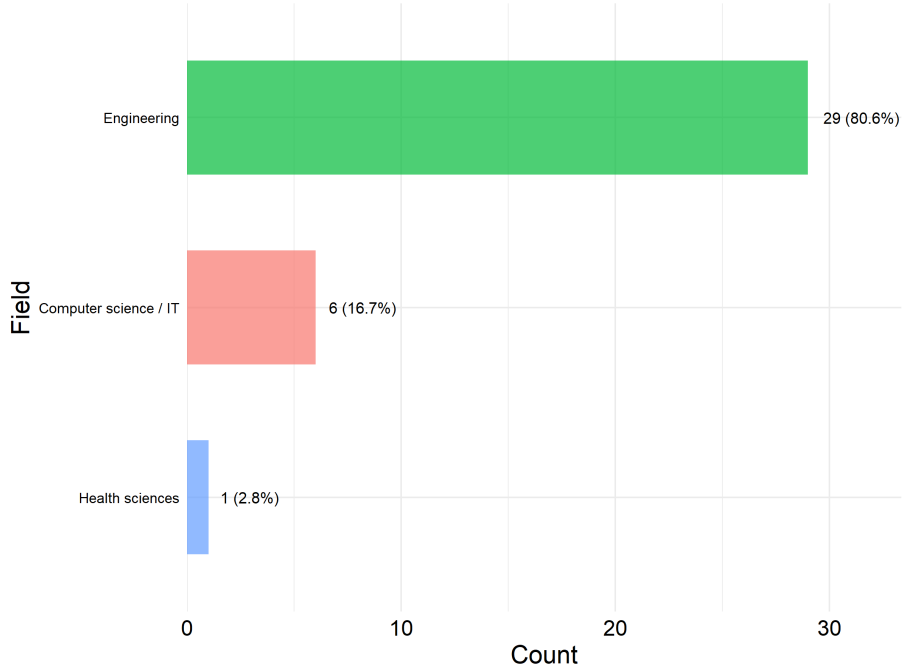


Figure 6.6: Distribution of participants' fields of study, highlighting the academic and disciplinary backgrounds represented in the experimental sample.

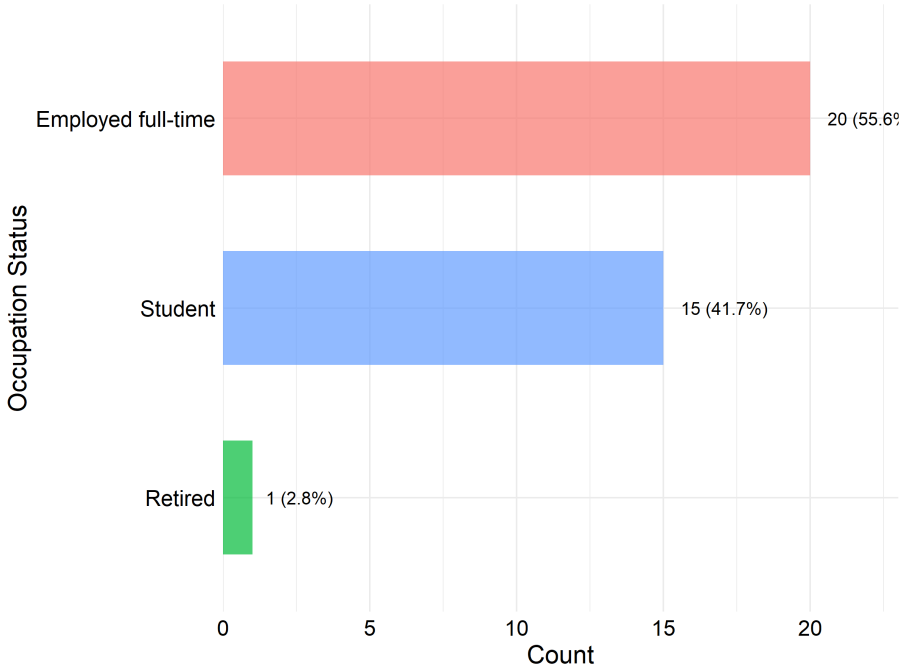


Figure 6.7: Occupational status of participants, providing insight into their current employment situation and professional engagement.

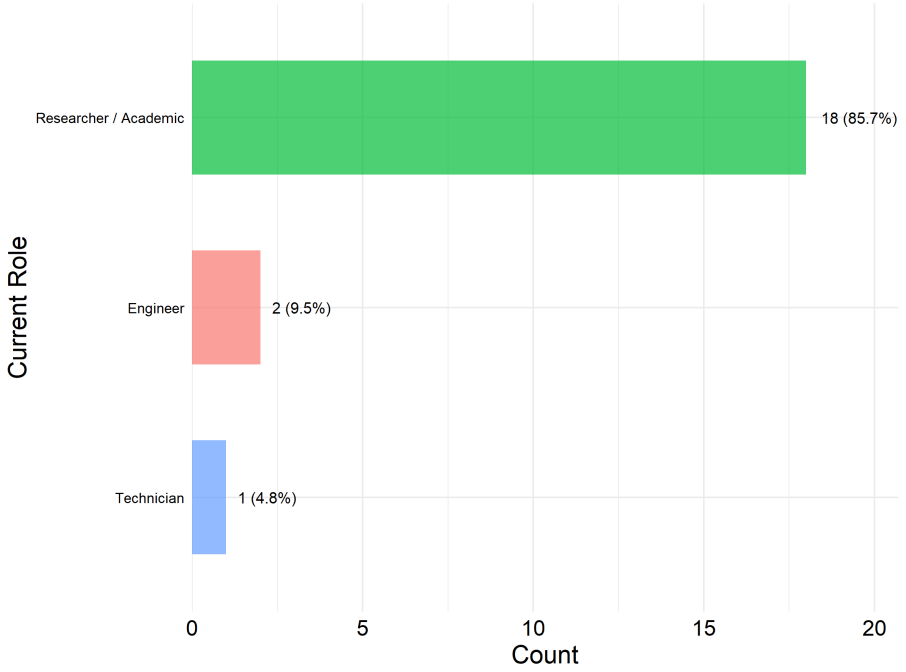


Figure 6.8: Current professional roles of participants, illustrating the positions held and functional responsibilities within their respective organizations.

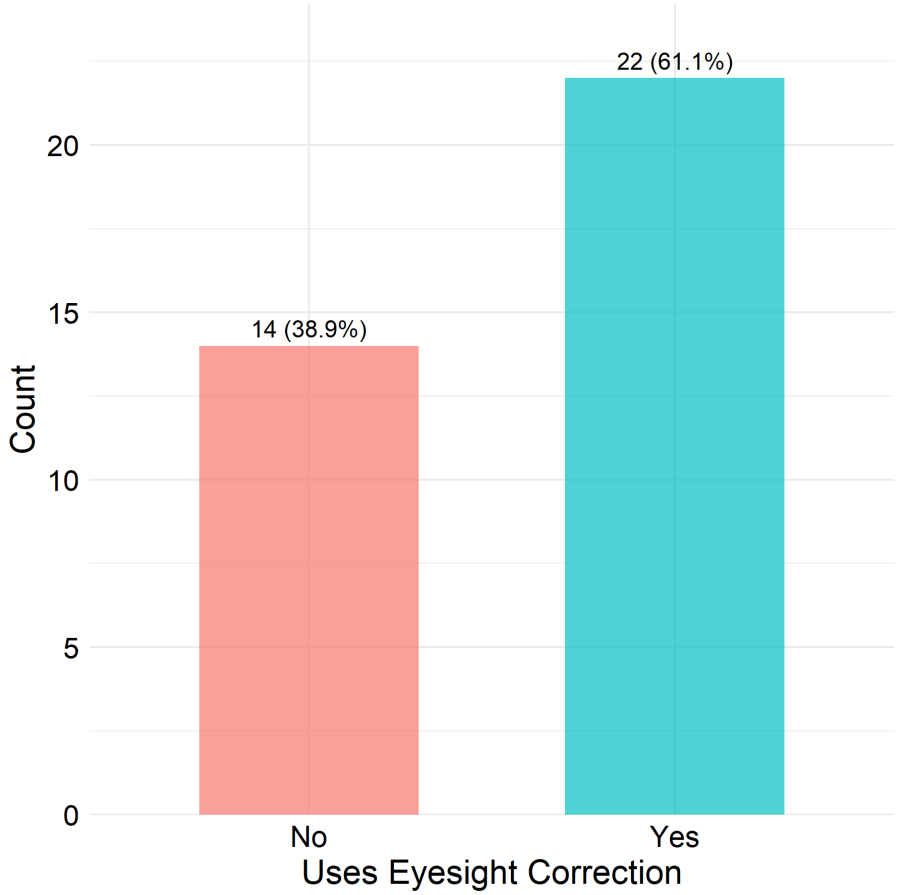


Figure 6.9: General use of eyesight correction among participants, indicating how many regularly wear glasses or contact lenses.

the AR interface, familiarization consisted of the in-application tutorial described in Section 6.4.2. For the touch screen, participants operated the interface through the same step-by-step procedure while a researcher explained each step at their side. In both cases, participants were encouraged to request clarification before proceeding.

Each session consisted of a complete run of the collaborative disassembly task. For each operation, participants reviewed the task instructions (video and text on the touch screen, or holographic animation in AR) and then executed the step, coordinating with the robot as required by the task type. Human-exclusive steps were performed independently; robot-exclusive steps required the participant to monitor and confirm robot completion; and collaborative steps involved coordinated manipulation between the participant and the UR5, with the participant handling assigned components while the robot managed its own. Participants were instructed to prioritize correct execution over speed, though task execution times were recorded for each operation.

Task progression in both interfaces was governed by an explicit confirmation flow. Participants signaled the start and completion of each operation through dedicated interface controls, which also triggered the scheduling algorithm to compute the next task and, for robot-involved operations, initiated or released robot motion. This interaction flow was identical in structure across

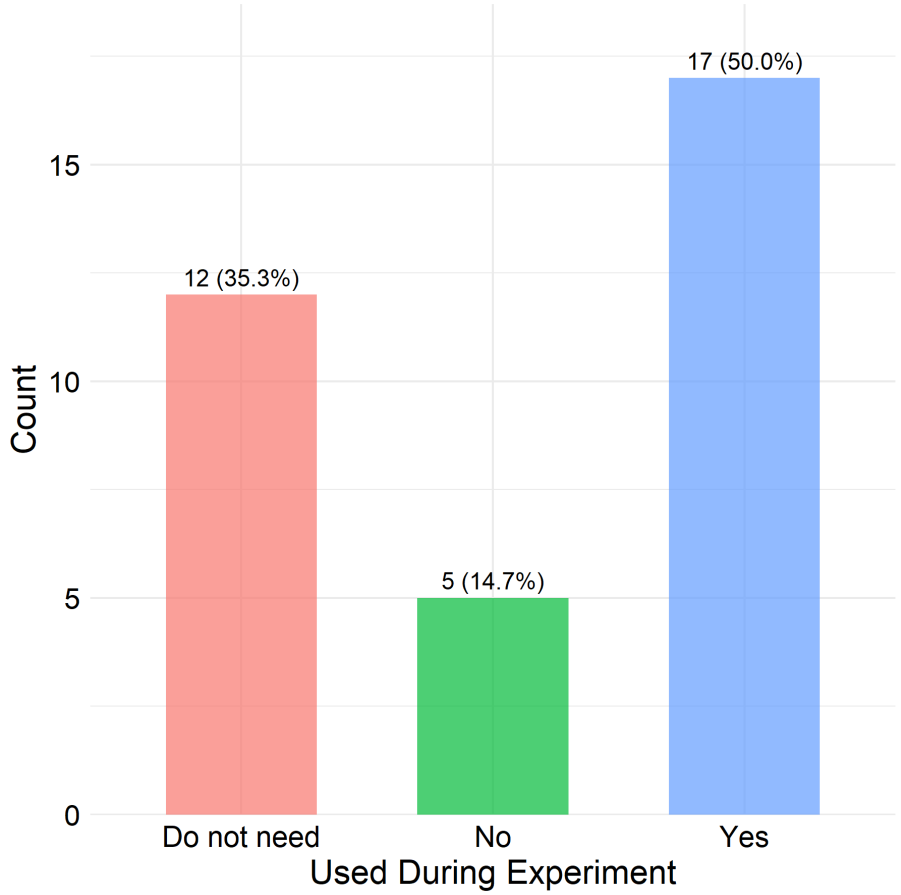


Figure 6.10: Number of participants who wore eyesight correction (glasses or contact lenses) during the experimental tasks, providing insight into visual accessibility during the study.

both interfaces, while the way in which confirmations were entered and acknowledged differed between devices (Section 6.5). The comparison therefore captures the full interaction package of each configuration, including its associated feedback mechanisms, confirmation demands, and progression rhythm, rather than isolating visualization alone.

Immediately after each session, participants completed the NASA-TLX, the SUS, and a custom post-trial questionnaire adapted from Lunding, Lystbæk, Feuchtner, *et al.* [26], covering perceived workload, usability, and additional evaluations of system features. An open comment field allowed optional qualitative feedback. After the second session, participants also took part in a brief semi-structured interview aimed at capturing additional insights and suggestions for improvement. The interview guide followed the topics of the interview used by Lunding, Lystbæk, Feuchtner, *et al.* [26] (clarity and usefulness of the instructions in each interface, differences between the interfaces and preference, awareness of the robot's actions and status, use of the robot movement preview, and the perceived collaboration with the robot), with follow-up questions when new topics emerged (Appendix A.5). Interviews lasted approximately ten minutes, were conducted in the participant's main language (Portuguese or English), and were audio-recorded. The complete session, comprising both conditions, the questionnaires, and the interview, lasted

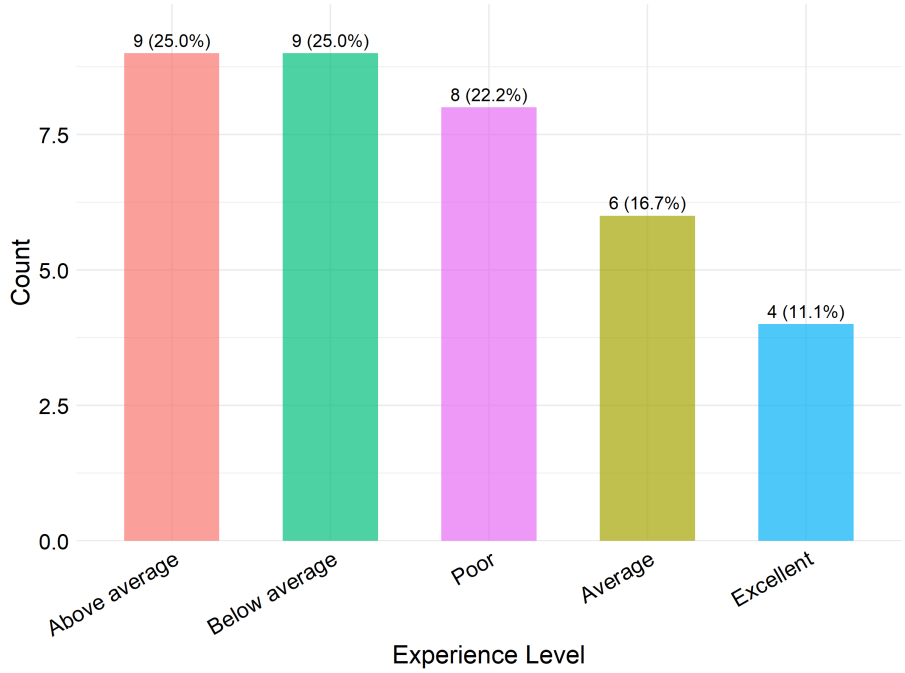


Figure 6.11: Participants' self-reported experience with AR/VR technologies, indicating familiarity levels prior to the experiment.

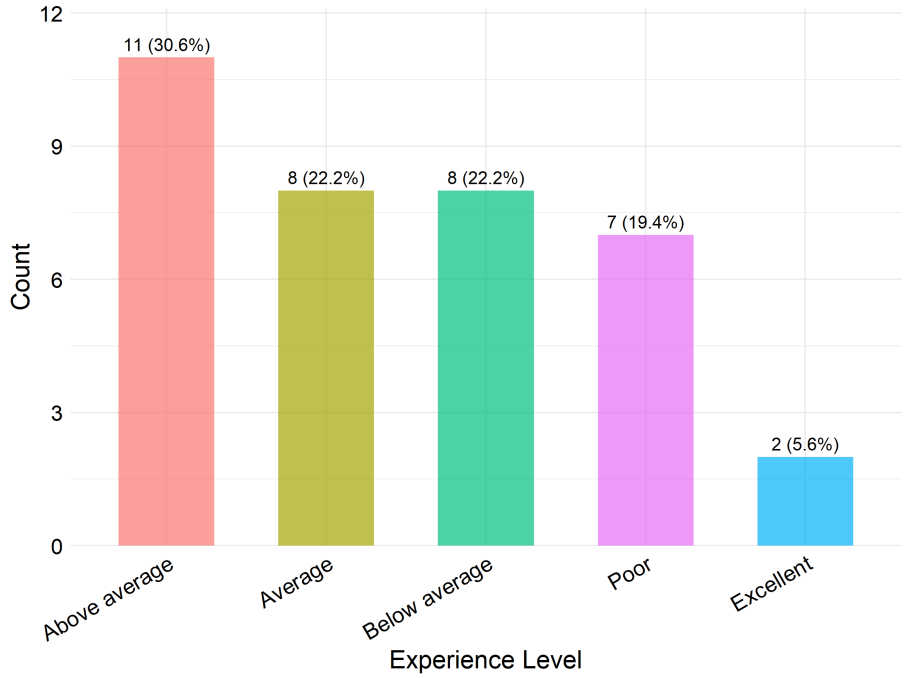


Figure 6.12: Participants' self-reported experience with disassembly tasks, reflecting prior exposure to similar hands-on technical activities.

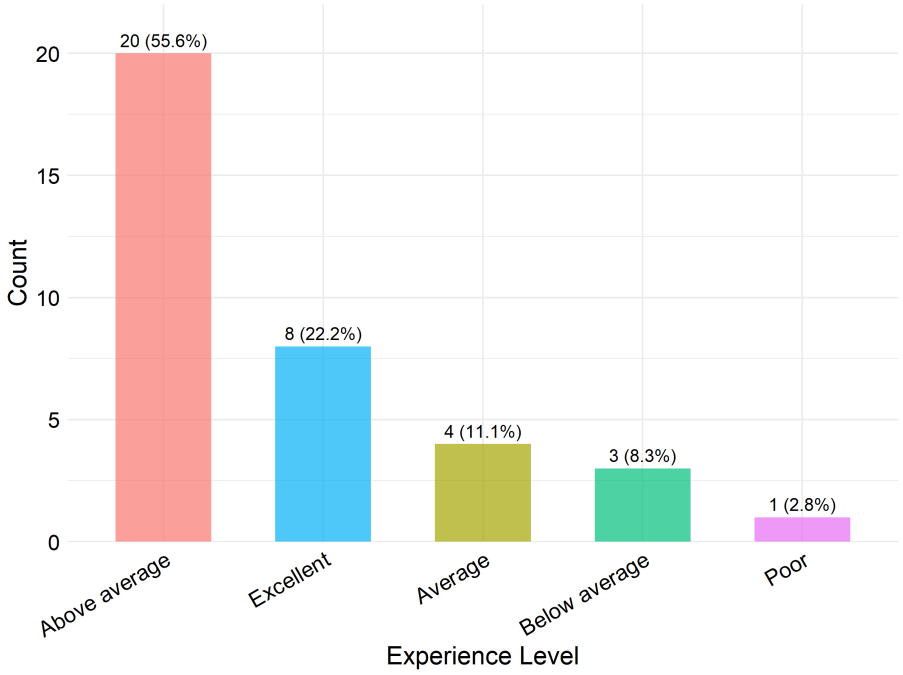


Figure 6.13: Participants' self-reported experience working with robots, highlighting prior interaction and familiarity with robotic systems in industrial or research settings.

approximately one hour per participant; the task itself lasted on average 7.5 and 7.8 minutes per condition with the touch screen and AR interface, respectively.

6.7.3 Dependent Measures and Statistical Analysis

The evaluation was designed to capture the effects of the interface configuration across multiple dimensions of HRI quality, spanning both performance- and interaction-oriented outcomes. Dependent measures were organized into two categories.

Quantitative measures included total execution time, defined as the sum of individual operation durations, and total procedure time, defined as the duration of the entire disassembly session. The number and type of errors committed during each session were also recorded. Errors were classified by type (forgetting to press a confirmation button, failing to remove a component, or proceeding before the robot had fully stopped) and by the operation in which they occurred, allowing analysis of whether error patterns differed between interfaces and across task types.

Qualitative measures were collected through four instruments. The NASA-TLX [187] assessed perceived cognitive workload across six subscales (mental demand, physical demand, temporal demand, performance, effort, and frustration), supplemented by an additional eye-fatigue item following Lunding, Lystbæk, Feuchtner, *et al.* [26], given the relevance of visual strain to prolonged HMD use. All items were rated on an 11-point scale from 0 (very low) to 10 (very high), except the performance item, whose anchors were presented in the more intuitive direction (0: failure; 10: perfect). Responses to this item were recoded $(10 - x)$ before analysis, so that, as in the original instrument, lower values indicate better perceived performance and all subscales

share the workload direction. The unweighted (raw) NASA-TLX score was computed as the mean of the six subscales [188]; a second score including the eye-fatigue item was also computed. The SUS [189] assessed overall perceived usability. A custom questionnaire, structured around three thematic blocks, captured dimensions not covered by standardized instruments. The General Questions block assessed the overall task experience, instruction clarity, perceived performance, and progress awareness; the Working with the Robot block assessed perceived collaboration, robot action awareness, sense of independence versus coordination, attention allocation, and work pace; and the AR Visualization block, completed only after the AR session, assessed the clarity of holographic instructions, the usefulness of the robot trajectory preview, and comprehension of robot intent. Semi-structured interviews conducted after the second session provided additional qualitative insight into participants' experiences, preferences, and suggestions for improvement. All instruments are reproduced in Appendix A.

The Working with the Robot and AR Visualization blocks represent the most directly HRI-relevant dimensions of the evaluation. While workload and usability capture the general interaction cost of each configuration, these blocks specifically target whether the interface supports the operator in understanding what the robot is doing, anticipating what it will do next, perceiving the collaboration as coordinated rather than parallel, and feeling aware of robot position and safety zones during shared task execution. These are the dimensions on which the two interfaces are most directly expected to differ (expectation E1, Section 6.1), and they provide the primary basis for interpreting the results in terms of interface effects on HRI quality.

6.7.3.1 Custom Questionnaire: Sources, Scales, and Reliability

All items of the custom questionnaire were based on the post-trial questionnaire of Lunding, Lystbæk, Feuchtner, *et al.* [26], who developed them to evaluate AR support in a human-robot assembly task comparable to the present one. The items were adapted to the present task by replacing references to the building task with references to the disassembly of the car alternator, and the original 7-point agreement scale was replaced by a 5-point Likert scale (1: strongly disagree; 5: strongly agree) to match the format of the SUS. Items of the original questionnaire that referred to features absent from the present system (task delegation through a playlist and direct control of the robot) were not used. The full item list is given in Appendix A.4. Four items are negatively worded and were reverse-coded before analysis ($x' = 6 - x$), so that higher scores always indicate a more favorable rating: losing track of the robot, paying more attention to the robot than to the task, and perceiving human and robot as working independently (Working with the Robot), and ambiguity of the AR instructions (AR Visualization). Block scores were computed as the mean of the item scores.

The items were chosen because they operationalize the constructs of Section 6.1 at the level of individual statements (e.g., awareness of the robot's next action for Level 3 SA, losing track of the robot for Level 1 SA, task-ownership clarity), with short wording suitable for repeated administration within a one-hour session. Established HRI instruments were considered but not adopted: the Situation Awareness Rating Technique [190] measures generic SA dimensions that

do not distinguish awareness of the robot from awareness of the task; the Godspeed questionnaire [191] and the Trust Perception Scale-HRI [192] target perceptions of the robot, which was identical in both conditions, rather than of the interface; and the human-robot fluency scales [172] overlap with the collaboration items but would have lengthened the questionnaire considerably. Using the items of Lunding, Lystbæk, Feuchtner, *et al.* [26] also allows a direct comparison with that study.

Internal consistency was assessed with Cronbach's α [193]. It was acceptable for the SUS ($\alpha = 0.80$ for both interfaces) and the raw NASA-TLX ($\alpha = 0.72$ touch screen; 0.70 AR), and moderate for the General Questions block (0.69; 0.59). It was low for the Working with the Robot block (0.48 touch screen; -0.01 AR) and the AR Visualization block (0.36), whose items address distinct facets (awareness, attention, pace, collaboration) rather than a single latent construct, and whose AR ratings clustered near the top of the scale (Table B.2). The block scores of the custom questionnaire are therefore treated as summary indices rather than as reliable unidimensional scales, and construct-specific conclusions are drawn from the individual items, analyzed with multiplicity control.

6.7.3.2 Statistical Analysis

Outcomes were organized into three tiers. The *primary outcomes*, one per measured construct, were the Working with the Robot index, the SUS score, the raw NASA-TLX score, the eye-fatigue item, the General Questions index, total execution time, total procedure time, and the number of errors. The *secondary outcomes* were the item-, subscale-, and operation-level analyses of these measures. The *exploratory analyses* comprised the comparisons between participant subgroups defined by prior AR/VR, disassembly, and robot experience and by the use of visual correction.

To account for the crossover design, each primary outcome was analyzed with a linear mixed model

$$y_{ij} = \mu + \tau_{I(i,j)} + \pi_j + \lambda_{s(i)} + u_i + \varepsilon_{ij}, \tag{6.1}$$

where y_{ij} is the outcome of participant i in session $j \in \{1,2\}$, τ is the effect of the interface used in that session, π the session (period) effect, λ the effect of the sequence $s(i)$ to which the participant was assigned, $u_i \sim \mathcal{N}(0, \sigma_u^2)$ a participant random intercept, and ε_{ij} the residual [194]. The interface effect (AR minus touch screen) is thus estimated within participants and adjusted for session; the session effect (second minus first session) captures practice, as well as fatigue or boredom, independently of the interface; and the sequence term captures differences between the two groups which, in a two-period crossover, are aliased with the interface-by-session interaction, i.e., with differential carryover. Models were fitted by restricted maximum likelihood, with Satterthwaite degrees of freedom [195], [196]. Because several outcomes are ordinal or skewed, the three effects were also tested with the nonparametric crossover procedure of Koch [197], i.e., Wilcoxon rank-sum tests between sequences on the period differences, on the interface differences, and on the participant totals, as a robustness check.

Secondary outcomes were analyzed with paired tests, as in the analysis of the primary outcomes without the session term. Normality of the paired differences was assessed with the Shapiro-

Wilk test and the D'Agostino skewness test at $\alpha = 0.05$; when both indicated normality, a paired t-test was used, otherwise the Wilcoxon signed-rank test. Effect sizes are reported as Cohen's d_z (mean difference divided by the standard deviation of the differences) with 95% confidence intervals. Where relevant, the session effect estimated with the mixed model is also reported. Error-type and error-location distributions were compared with Fisher's exact test, given the small cell counts.

Multiplicity was controlled with the Holm procedure [198], applied across the eight primary outcomes for each model term, and within each family of secondary tests: the six Working with the Robot items, the four General Questions items, the seven NASA-TLX items (six subscales and eye fatigue), the ten SUS items, the nine operations, the five error types, and the nine error locations. Unadjusted and Holm-adjusted p-values (p_{Holm}) are both reported, and significance statements refer to the adjusted values. Exploratory subgroup comparisons (Kruskal-Wallis tests between experience groups, followed by within-group paired tests) were not corrected; they are reported in Appendix B and interpreted descriptively. All tests were two-sided at $\alpha = 0.05$ and were performed in R.

6.7.3.3 Interview Analysis

The interview recordings were transcribed verbatim by the author. The transcripts were then analyzed thematically: all answers were read, recurring topics were identified, both those anticipated by the interview guide and those that emerged during the interviews, and for each topic the participants who mentioned it were recorded. These counts were used to describe how widely each topic was shared, and the interview results are reported in aggregate form alongside the corresponding quantitative results. Statements given in Portuguese were translated into English for reporting. The analysis was conducted by a single researcher, and its limitations are discussed in Section 6.9.

6.7.4 Interfaces Results

The results are structured into five blocks. The first summarizes the primary outcomes and the order effects estimated with the crossover model. The second focuses on outcomes most directly related to HRI quality, grouping the Working with the Robot and AR Visualization questionnaire results. The third addresses workload and usability, grouping eye fatigue, NASA-TLX, SUS, and General Task Experience results as measures of interaction cost. The fourth presents objective performance outcomes, grouping error analysis, execution times, and practice and order effects. The fifth reports exploratory subgroup analyses, which are secondary and not part of the primary comparisons. Interview data are embedded within each block, rather than presented separately, to provide immediate qualitative context for the corresponding quantitative results. Improvement suggestions gathered from participants and experts are discussed in Section 6.8, where they are interpreted as design implications of the results.

6.7.4.1 Overview of Primary Outcomes and Order Effects

Table 6.4 summarizes the eight primary outcomes and the interface and session effects estimated with the crossover model. After Holm correction, the AR configuration differed significantly from the touch screen on four outcomes: it produced a higher Working with the Robot index, a lower SUS score, higher eye fatigue, and longer total execution time. No significant interface effect was found for the raw NASA-TLX, the General Questions index, total procedure time, or the number of errors. The nonparametric crossover tests led to the same conclusions for all eight outcomes.

Session effects were substantial for time-related outcomes. Irrespective of the interface, participants were 51.2 s faster in total execution time and 119.5 s faster in total procedure time in their second session, an effect on execution time of the same magnitude as the interface effect. Raw NASA-TLX workload (-0.49 points, $p = 0.010$, $p_{\text{Holm}} = 0.063$) and the number of errors (-0.47, $p = 0.054$) also tended to decrease in the second session. The sequence term was not significant for any outcome after correction; nominal sequence effects for eye fatigue ($p = 0.033$) and errors ($p = 0.020$) reflected overall differences between the two groups, with the interface difference pointing in the same direction in both sequences (full model and nonparametric results in Table B.1; see the eye-fatigue and error analyses in Sections 6.7.4.3 and 6.7.4.4).

The exploratory screening of participant subgroups (Kruskal-Wallis tests) found no systematic differences in any outcome by prior AR/VR experience, disassembly experience, robot experience, or use of visual correction, with two isolated exceptions: the raw NASA-TLX score with the touch screen differed between disassembly-experience groups ($p = 0.040$), and the General Questions index showed borderline differences between robot-experience groups ($p = 0.065$ and $p = 0.096$). These results and the follow-up subgroup analyses are reported in Appendix B.2 and summarized in Section 6.7.4.5.

Table 6.4: Primary outcomes: descriptive statistics and crossover mixed-model estimates of the interface effect (AR minus touch screen) and the session effect (second minus first session). Linear mixed model with interface, session, and sequence as fixed effects and a participant random intercept (df = 34). Holm correction across the eight outcomes, separately for each term. Sequence effects were non-significant after correction for all outcomes (unadjusted p from 0.020 to 0.860).

Outcome	Scale	TS, M (SD)	AR, M (SD)	Interface Δ [95% CI]	p (p_{Holm})	Session Δ [95% CI]	p (p_{Holm})
Working with the Robot index	1–5	3.76 (0.53)	4.01 (0.38)	0.25 [0.09, 0.41]	0.003 (0.014)	0.07 [−0.08, 0.23]	0.347 (1.000)
SUS	0–100	83.3 (11.6)	74.7 (15.0)	−8.68 [−13.47, −3.89]	<0.001 (0.005)	1.18 [−3.61, 5.97]	0.620 (1.000)
NASA-TLX (raw)	0–10	2.06 (1.36)	2.34 (1.27)	0.29 [−0.08, 0.65]	0.122 (0.488)	−0.49 [−0.86, −0.12]	0.010 (0.063)
Eye fatigue	0–10	1.22 (1.74)	2.72 (2.42)	1.50 [0.87, 2.13]	<0.001 (<0.001)	0.22 [−0.41, 0.85]	0.478 (1.000)
General Questions index	1–5	4.19 (0.62)	4.35 (0.56)	0.17 [−0.09, 0.42]	0.188 (0.563)	0.04 [−0.21, 0.29]	0.739 (1.000)
Total execution time	s	278.0 (63.5)	327.3 (81.5)	49.3 [26.5, 72.0]	<0.001 (<0.001)	−51.2 [−73.9, −28.4]	<0.001 (<0.001)
Total procedure time	s	449.0 (125.1)	465.8 (111.5)	16.7 [−22.6, 56.1]	0.393 (0.786)	−119.5 [−158.9, −80.2]	<0.001 (<0.001)
Errors per participant	count	0.78 (1.15)	0.92 (0.97)	0.14 [−0.34, 0.62]	0.560 (0.786)	−0.47 [−0.95, 0.01]	0.054 (0.268)

6.7.4.2 Human-Robot Interaction Outcomes

The most distinctive results of this study concern how the interface configuration shaped the operator's perceived experience of collaborating with the robot. Across both the custom questionnaire

and the interview data, the AR interface consistently produced stronger scores on dimensions directly related to robot awareness, with weaker evidence for perceived collaboration.

Perceived Collaboration and Robot Awareness

In the Working with the Robot block, the items referring to losing track of the robot, attending more to the robot than to the task, and perceiving the human and robot as working independently were reverse-coded so that higher scores indicated better perceived collaboration and robot awareness.

The Working with the Robot index was significantly higher with the AR interface (mean = 4.01, SD = 0.38) than with the touch screen (mean = 3.76, SD = 0.53), with a difference of 0.25 points (95% CI [0.09, 0.41], $p_{\text{Holm}} = 0.014$, $d_z = 0.54$) and no session effect (Table 6.4 and Fig. 6.14). Given the low internal consistency of the block (Section 6.7.3.1), the item-level results in Table 6.5 carry the construct-specific interpretation. Awareness of the robot's next action was the only item that differed significantly after Holm correction ($p = 0.005$, $p_{\text{Holm}} = 0.031$, $d_z = 0.53$). This effect remained when session was taken into account (mixed model: $+0.61$ points, $p = 0.002$); awareness also increased from the first to the second session ($+0.44$, $p = 0.018$), independently of the interface. Not losing track of the robot and feeling that one was collaborating with the robot showed small effects in favor of AR ($d_z = 0.34$ for both) that did not reach significance ($p = 0.052$ and $p = 0.053$) and did not survive correction. Work pace, attention allocation, and perceived independence did not differ between interfaces. The mean scores for each item are presented in Fig. 6.15; reverse-coded items are interpreted so that higher scores correspond to a lower perceived impact of the negatively worded statement.

Table 6.5: Working with the Robot items: paired comparisons between interfaces (AR minus touch screen). Item numbering as in Appendix A.4.2. W: Wilcoxon signed-rank test ($n = 36$). †Reverse-coded item.

Item	Construct	Test	p	p_{Holm}	d_z [95% CI]
1	Awareness of next action	W	0.005	0.031	0.53 [0.18, 0.88]
2	Not losing track†	W	0.052	0.259	0.34 [0.01, 0.68]
3	Pace influence	W	0.900	0.903	−0.02 [−0.35, 0.30]
4	Attention on task†	W	0.452	0.903	0.11 [−0.22, 0.43]
5	Collaboration	W	0.053	0.259	0.34 [0.00, 0.68]
6	Not working independently†	W	0.132	0.396	0.28 [−0.06, 0.61]

Participants thus reported greater awareness of the robot's upcoming actions when using the AR glasses, together with a tendency toward a stronger sense of collaboration and toward losing track of the robot less often. The comparable work pace scores indicate that these perceptions did not translate into a difference in perceived task rhythm.

The interview data are consistent with these findings. The majority of participants reported that the AR interface allowed them to see precisely where the robot would move and stop in the

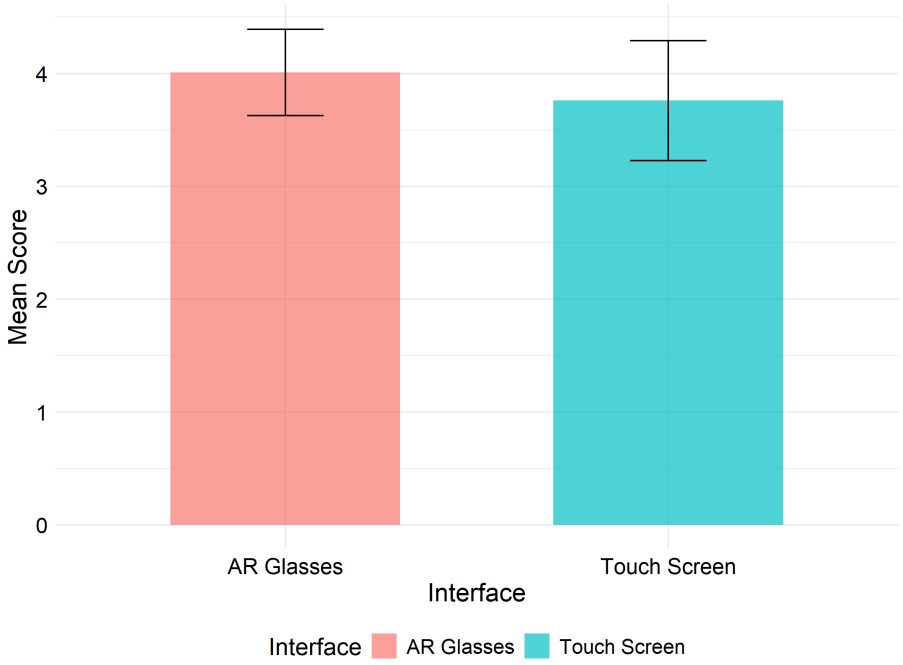


Figure 6.14: Mean scores for the "Working with the Robot" section of the custom questionnaire, grouped by interface type. This analysis highlights how participants' perceptions of collaboration and interaction with the robot vary depending on the interface used.

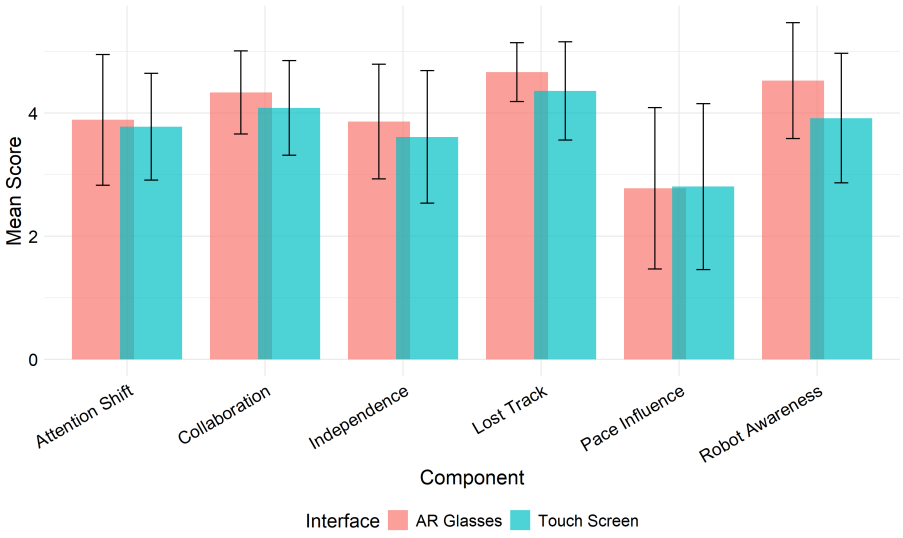


Figure 6.15: Mean scores for individual components of the "Working with the Robot" section of the custom questionnaire, grouped by interface type. This breakdown highlights which aspects of robot collaboration and interaction are most influenced by the interface used.

real workspace, rather than relying solely on video representations on the touch screen. Many noted that this visualization increased their sense of safety and confidence when working alongside the robot, and experts echoed these observations, highlighting the capacity of AR to convey robot trajectories and safety zones. Conversely, a subset of participants reported reduced spatial awareness of robot movements when using the touch screen, making it more difficult to anticipate upcoming robot actions.

Regarding the perceived utility of the robot, participants offered a consistent view across both interface conditions. No participant reported fear or significant stress during the collaborative task. Several participants noted that the disassembly task, given the simplified 3D-printed model, could have been completed faster by a human working alone, while others emphasized that the robot was particularly useful during collaborative operations, specifically in holding larger components while the operator used both hands to remove smaller parts. Ergonomic benefits of robot assistance were also noted, with several participants suggesting these would be more pronounced when handling heavier or real-world components.

AR Visualization Quality

In the AR Visualization block, the item "The AR instructions for disassembling the car alternator were ambiguous or unclear" was reverse-coded, as this item has a negative connotation.

Ratings did not differ between groups with different prior experience (Table B.4) or between the two sequences. Overall, the AR Visualization questionnaire yielded a mean score of 4.47 (SD = 0.453). At the item level, participants agreed that previewing the robot's movement overlaid on the real world was useful (mean = 4.61, SD = 0.60) and that it made the robot's intentions clear (mean = 4.75, SD = 0.44), and rated the AR instructions as unambiguous (mean = 4.47, SD = 0.81); agreement that the visualizations helped them understand the robot's status was lower (mean = 4.03, SD = 1.08), as shown in Fig. 6.16. Because this block was administered only after the AR session, it has no touch screen counterpart; these ratings are therefore descriptive and, given their proximity to the top of the scale, should be read as absolute acceptance of the AR features rather than as evidence of superiority over the touch screen.

These results are consistent with the interview feedback on instruction clarity and task understanding. The majority of participants reported that instructions were clearer and more intuitive when delivered through the AR interface, particularly in collaborative tasks. Several indicated that AR facilitated a deeper understanding of the disassembly process compared to relying solely on instructional videos. Experts described AR as superior for visualizing instructions within three-dimensional space, promoting more intuitive task comprehension. The color-coded task allocation, green for operator and red for robot, was also widely appreciated as enabling rapid identification of task ownership between human and robot.

A substantial proportion of participants, as well as all participating experts, expressed a preference for the AR interface in training and onboarding contexts, describing it as more immersive and supportive of learning, particularly in heterogeneous production environments. Several participants also believed that the spatial understanding acquired through AR carried over more ef-

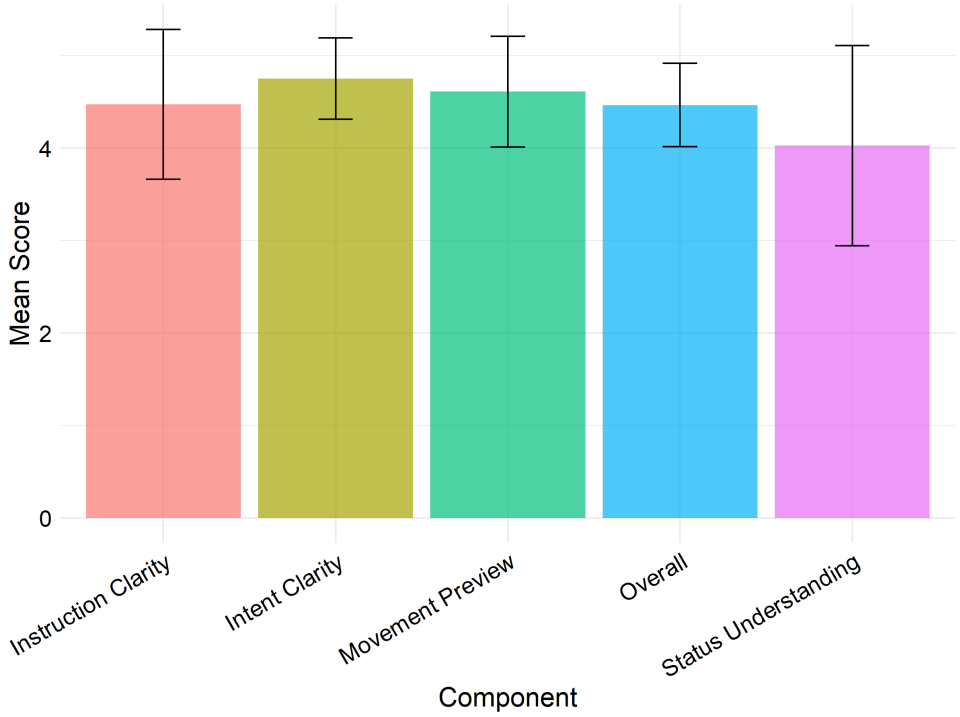


Figure 6.16: Mean scores for individual components of the "AR Visualization" section of the custom questionnaire, grouped by interface type. This visualization highlights which aspects of AR presentation were perceived most positively or negatively by participants.

fectively to subsequent touch screen use than the reverse direction; this self-reported asymmetry is examined against the performance data in Section 6.7.4.4.3.

6.7.4.3 Workload and Usability

While the AR interface produced stronger HRI outcomes, it also imposed higher interaction costs compared to the touch screen. This section presents the eye-fatigue, workload, usability, and general task experience results that characterize this trade-off.

Eye-fatigue

Participants reported significantly greater eye fatigue with the AR glasses (mean = 2.72, SD = 2.42) than with the touch screen (mean = 1.22, SD = 1.74), with a difference of 1.50 points on the 0–10 scale (95% CI [0.87, 2.13], $p_{\text{Holm}} < 0.001$, $d_z = 0.81$), as shown in Fig. 6.17. Eye fatigue did not change between sessions ($p = 0.478$) and did not differ between participants who did or did not use visual correction (Kruskal-Wallis, AR glasses $p = 0.284$; touch screen $p = 0.253$; Table B.5). Participants who started with the touch screen reported higher eye fatigue in both conditions than those who started with AR (touch screen: 1.78 vs. 0.67; AR: 3.50 vs. 1.94), which produced the nominal sequence effect; the AR condition was rated as more fatiguing in both sequences.

This finding was reflected in participant comments. A substantial number of participants expressed concerns regarding eye fatigue and visual discomfort during prolonged AR use. Several

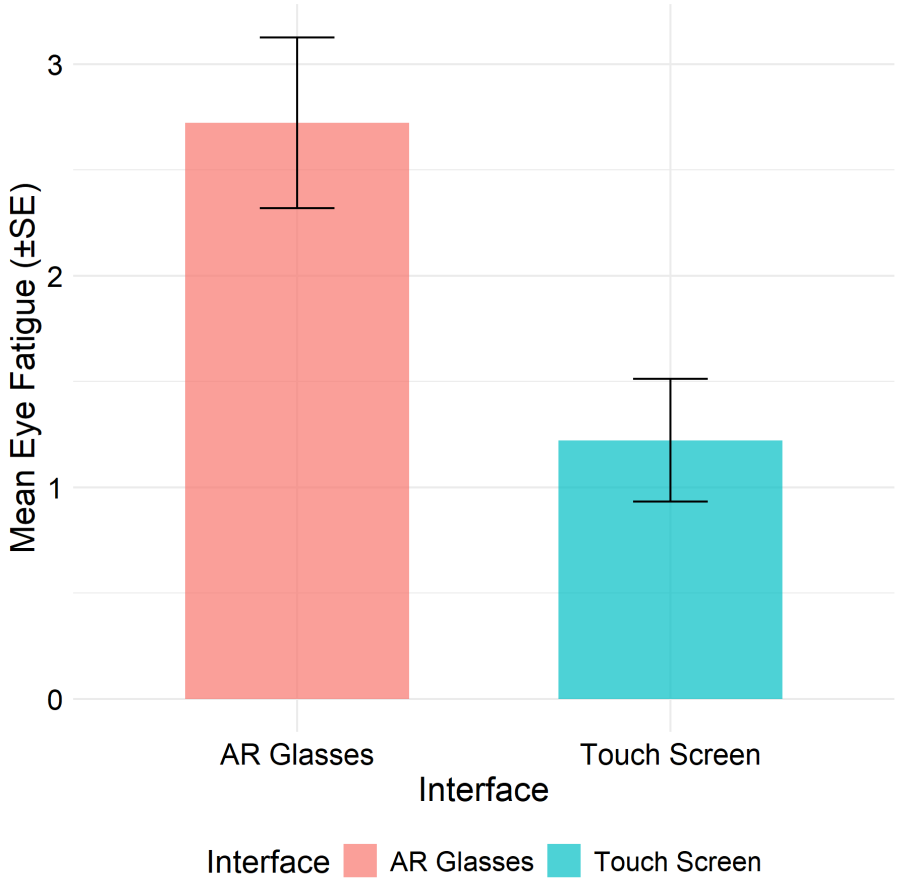


Figure 6.17: Mean eye-fatigue reported by participants for each interface, illustrating the visual comfort differences between the AR and touch screen conditions.

attributed this to holographic overlays described as excessively opaque and featuring overly intense colors, which hindered visibility and made it more difficult to accurately perceive and manipulate physical components. The limited field of view of the HL2 was also raised by several participants, who reported frequently needing to turn their heads to access information outside the visible area. A smaller number of participants noted that the touch screen was less visually demanding and less fatiguing during prolonged use.

NASA-Task Load Index

The raw NASA-TLX score did not differ significantly between interfaces (AR minus touch screen: 0.29 points, 95% CI [−0.08, 0.65], $p = 0.122$), but decreased from the first to the second session (−0.49 points, $p = 0.010$, $p_{\text{Holm}} = 0.063$). When the eye-fatigue item was included, the AR condition showed a higher overall workload (0.46 points, 95% CI [0.13, 0.79], $p = 0.007$), again with a lower workload in the second session (−0.39, $p = 0.021$). In all cases, the mean scores indicate a medium level of perceived workload for both interfaces, according to the workload classification proposed by Prabaswari, Basumerda, and Utomo [199], as shown in Fig. 6.18.

At the subscale level (Table B.15), eye fatigue was the only item that differed significantly after

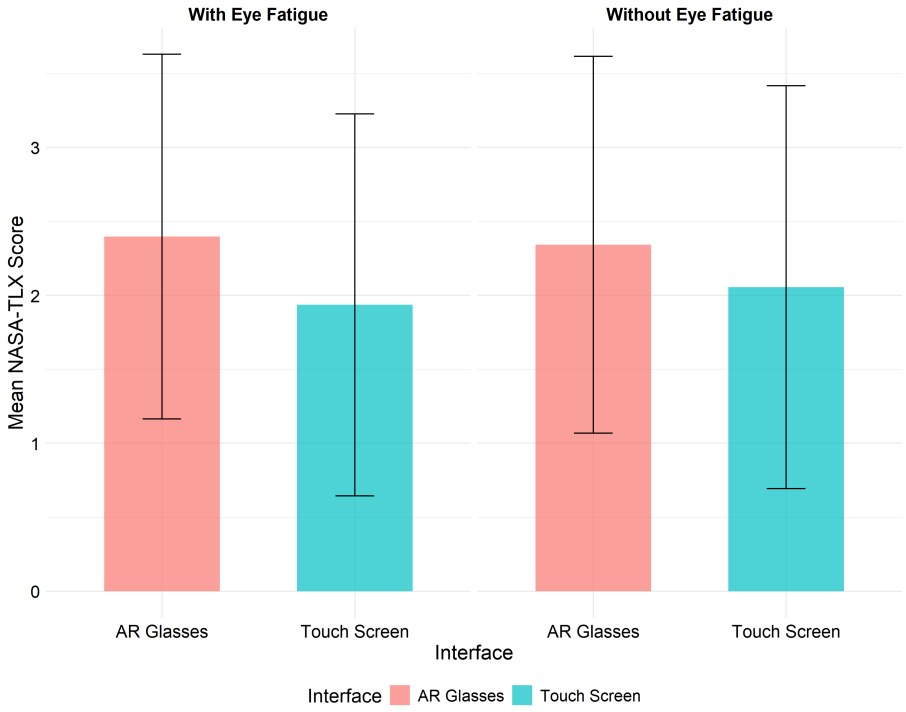


Figure 6.18: Average NASA-TLX scores for each interface, showing the overall perceived workload across participants and highlighting differences in cognitive, physical, and temporal demands between the AR and touch screen conditions.

correction ($p_{\text{Holm}} < 0.001$). Frustration ($p = 0.051$, $d_z = 0.28$) and physical demand ($p = 0.075$, $d_z = 0.33$) were descriptively higher with AR but not significant, and mental demand, temporal demand, performance, and effort did not differ. Mental demand showed the strongest decrease between sessions (-0.97 points, $p < 0.001$), consistent with the reduction of task novelty in the second session. Mean subscale scores are shown in Fig. 6.19. The AR glasses were rated higher on all subscales except temporal demand and performance. The performance scale is inverted; lower scores indicate better perceived performance.

The descriptively higher frustration is consistent with participant feedback. A substantial number of participants reported that the HL2 interaction methods, including pinch gestures, close-range virtual button presses, and text input, were inconsistent and characterized by a steep learning curve. The absence of clear button feedback increased uncertainty about whether inputs had been successfully registered. The virtual keyboard was a further source of difficulty, frequently appearing in unexpected locations and lacking a clear visual connection between typing gestures and text entry. A smaller number of participants also reported physical discomfort related to device weight and extended arm postures during AR-based interactions.

System Usability Scale

Participants rated the touch screen as significantly more usable than the AR glasses (difference of -8.68 points, 95% CI $[-13.47, -3.89]$, $p_{\text{Holm}} = 0.005$, $d_z = -0.62$), with no session effect ($p =$

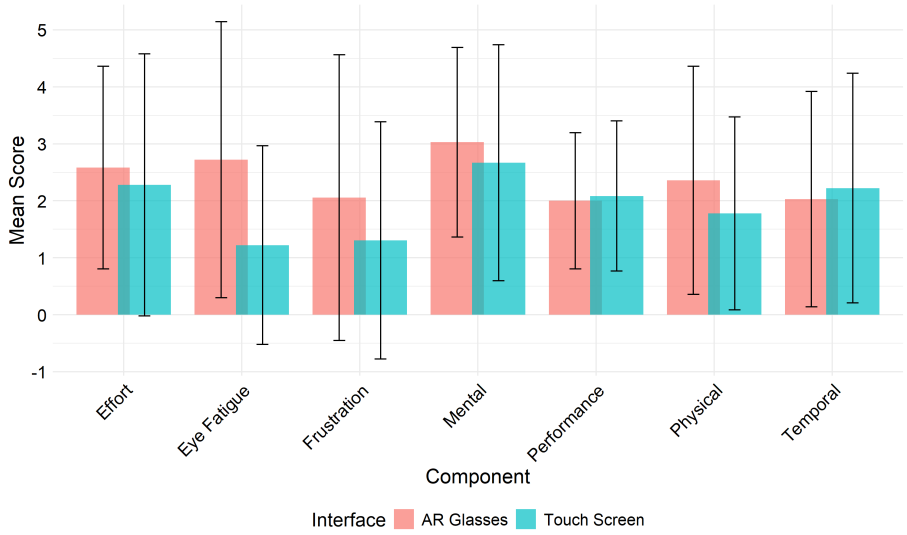


Figure 6.19: Mean scores of individual NASA-TLX workload components for each interface, illustrating differences in mental, physical, and temporal demands, as well as performance, effort, and frustration levels between the AR and touch screen conditions.

0.620). The mean SUS score was 74.7 (SD = 15.0) for the AR glasses and 83.3 (SD = 11.6) for the touch screen, as illustrated in Fig. 6.20. According to the usability adjective ratings proposed by Bangor, Kortum, and Miller [200], both interfaces fall within the "good" usability range; however, the AR interface is closer to the "OK" threshold, while the touch screen approaches the "excellent" category.

At the item level (Table B.16 and Fig. 6.21), two items differed after Holm correction: participants agreed more strongly that they would need the support of a technical person to use the AR system (item 4, $p_{\text{Holm}} = 0.039$) and less strongly that most people would learn to use it very quickly (item 7, $p_{\text{Holm}} = 0.039$). Items on frequent use (item 1), ease of use (item 3), and cumbersomeness (item 8) favored the touch screen at the nominal level ($p < 0.05$) but did not survive correction. The differences therefore concentrate on learnability and the need for support, rather than on the integration or consistency of the system functions.

The usability gap between interfaces was consistently reflected in participant and expert feedback. The majority of participants described the touch screen as easier and more intuitive, largely attributing this to their familiarity with everyday touch-based devices. Several participants and experts emphasized the value of haptic feedback, noting that tactile confirmation provided clear assurance that inputs had been registered and reinforced a sense of control during task execution. Some participants also appreciated that the touch screen allowed simultaneous access to all relevant information without requiring head movement. The majority of participants expressed a preference for the touch screen for continuous, day-to-day operation following the training phase. In contrast, 50% of participants preferred the AR interface for training and onboarding, and several identified a hybrid approach combining touch screen input with AR-based visualization as a promising direction.

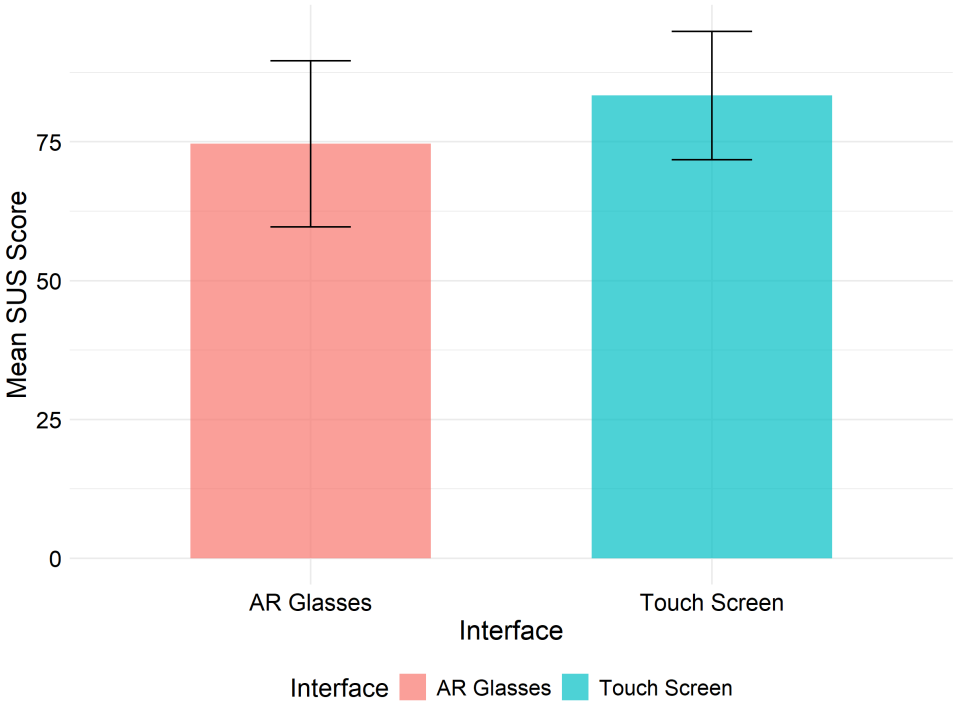


Figure 6.20: Mean SUS scores for each interface, providing a group-level comparison of perceived usability across the evaluated systems.

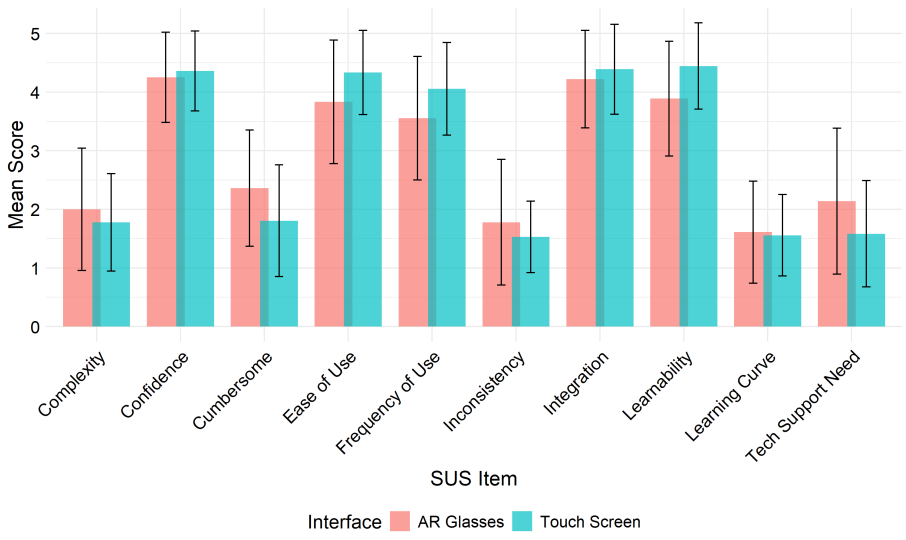


Figure 6.21: Mean scores of individual SUS components for each interface, illustrating a detailed comparison of usability aspects across the evaluated systems at the group level.

General Task Experience

The General Questions index did not differ significantly between interfaces (0.17 points, 95% CI $[-0.09, 0.42]$, $p = 0.188$), and no item differed after correction (Table B.17). The AR condition showed slightly higher mean scores on all four items (feeling in charge, awareness of overall progress, perceived performance, and ease of following the instructions), without evidence of a session effect, as illustrated in Fig. 6.22 and Fig. 6.23.

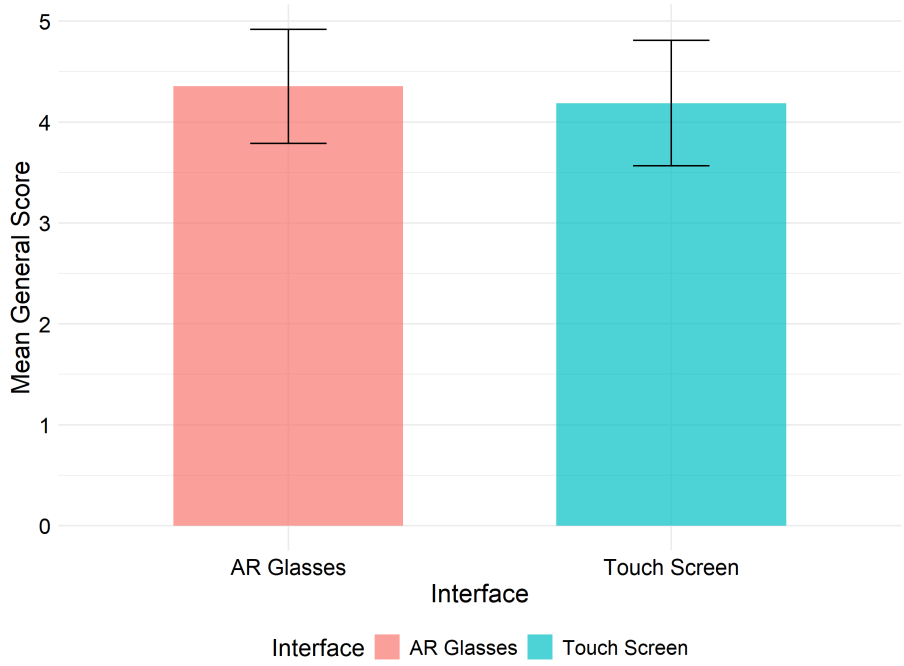


Figure 6.22: Mean scores of the general questions from the custom questionnaire, comparing overall user perceptions across interfaces at the group level.

6.7.4.4 Performance Indicators

Error Analysis

The actions considered as errors in this section included forgetting to press a system button ("INI", "END", or "Continue"); failing to remove a component during the disassembly process; and an early finish, defined as proceeding to the next step (pressing the "END" button) before the robot had completely stopped.

In total, 33 errors were recorded with the AR glasses and 28 with the touch screen, corresponding to a mean of 0.917 errors per participant (SD = 0.967) for the AR glasses and 0.778 errors per participant (SD = 1.15) for the touch screen. Of the 36 participants, 21 committed at least one error with the AR glasses and 19 with the touch screen; only 6 committed no errors with either interface (Fig. 6.24, Fig. 6.25, and Fig. 6.26). The crossover model revealed no interface effect on the number of errors (0.14, 95% CI $[-0.34, 0.62]$, $p = 0.560$), but errors tended to be more frequent in the first session (-0.47 in the second session, $p = 0.054$). Participants who started

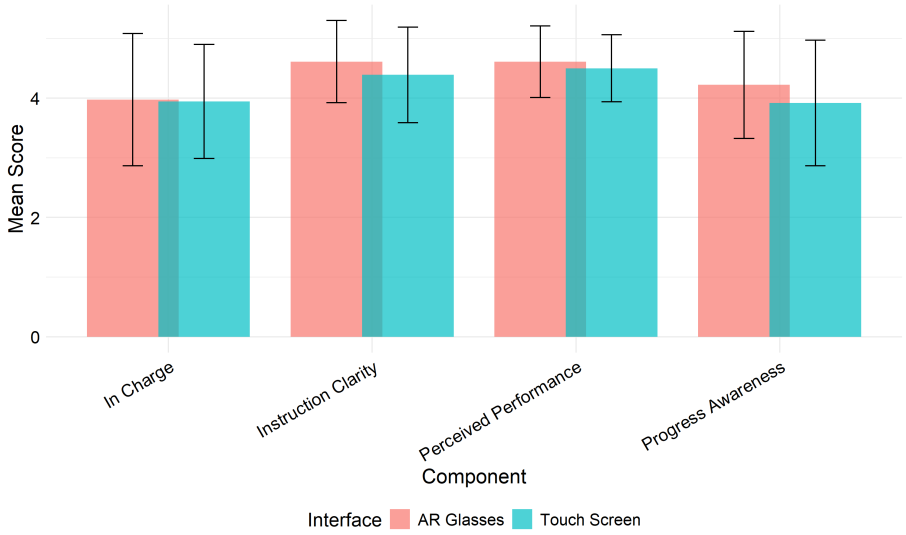


Figure 6.23: Mean scores of individual components for the general questions of the custom questionnaire, highlighting detailed user feedback and differences between interfaces at the group level.

with AR made more errors across both sessions than those who started with the touch screen (AR: 1.44 vs. 0.39; touch screen: 0.83 vs. 0.72), which accounts for the nominal sequence effect; the highest error rate occurred in first-session use of the AR interface.

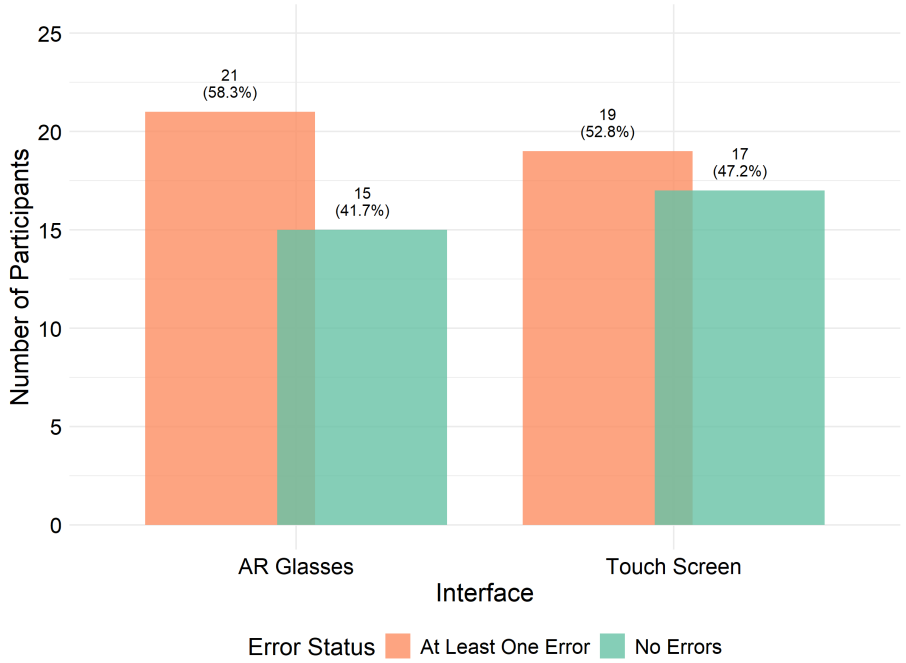


Figure 6.24: Distribution of participants who made errors versus those who did not, grouped by interface type. This highlights the relative effectiveness of each interface in minimizing user errors during the task.

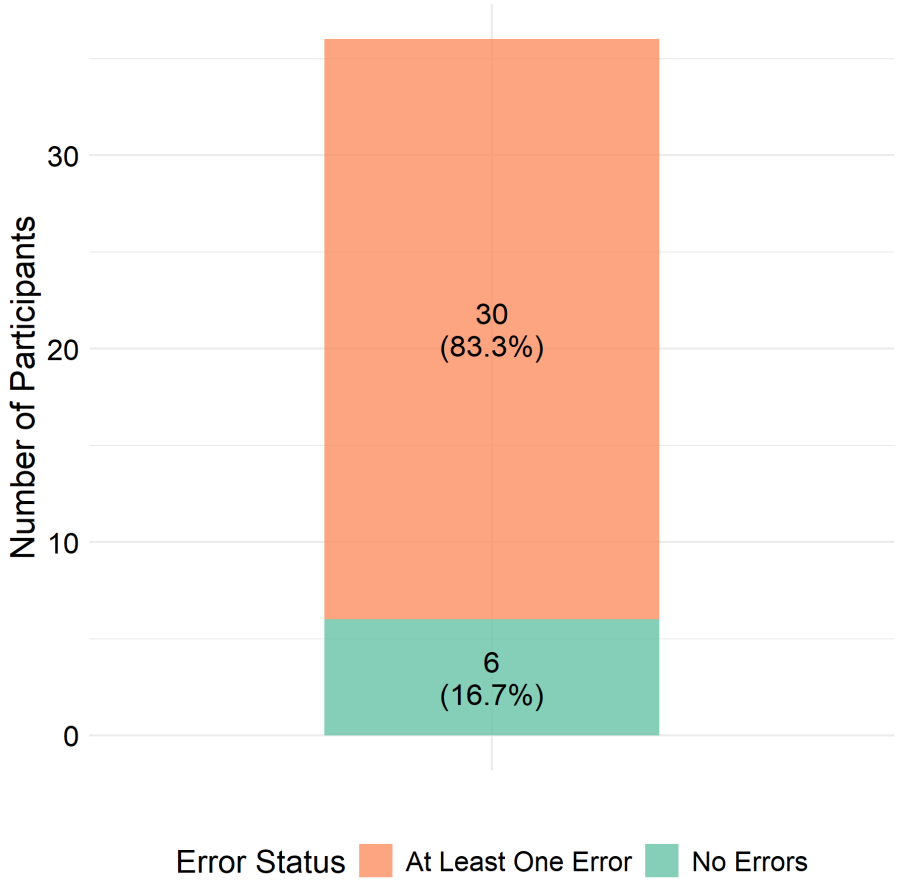


Figure 6.25: Total error occurrences aggregated across all participants, providing an overview of common mistakes and the frequency of errors observed during the experiment.

Critically, 51 of the 61 recorded errors across both conditions were interaction errors (forgetting to press a required button) rather than task execution errors such as failing to remove a component. The remaining errors corresponded to early finish events (5 cases) and forgotten components (5 cases). When stratified by error type, 63.6% of the errors in the AR condition involved forgetting the "INI" button, 18.2% forgetting the "Continue" button, 9.1% forgetting the "END" button, and 9.1% forgetting to remove a component. In the touch screen condition, 60.7% of errors involved forgetting the "INI" button, 17.9% early finishes, 14.3% forgetting the "Continue" button, and 7.1% forgetting to remove a component, as shown in Fig. 6.27, Fig. 6.28, and Fig. 6.29.

Fisher's exact tests revealed no significant differences in error type or error location between interfaces after Holm correction (Table B.19 and Table B.20), and the global chi-square test did not indicate that the distribution of errors across operations differed between interfaces (Table B.21). The only nominal difference concerned the early-finish error (0 with AR vs. 5 with the touch screen, $p = 0.017$, $p_{\text{Holm}} = 0.083$), which was driven by a single participant.

Several participants and experts noted that the initial button sequence required to trigger the scheduling algorithm was not intuitive. Both the absence of clear visual feedback indicating when

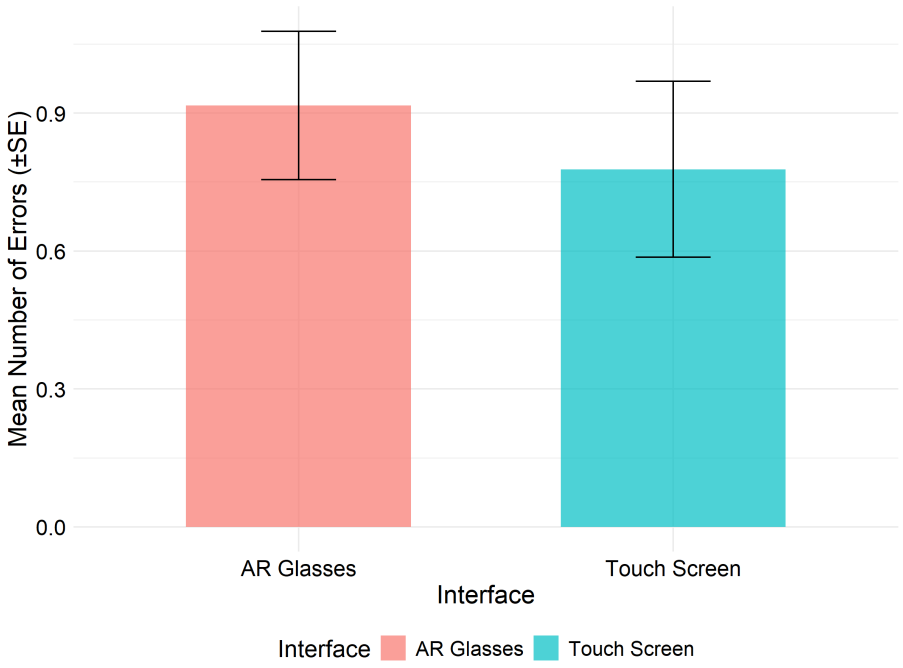


Figure 6.26: Average number of errors per participant for each interface type, illustrating the comparative impact of interface design on task accuracy.

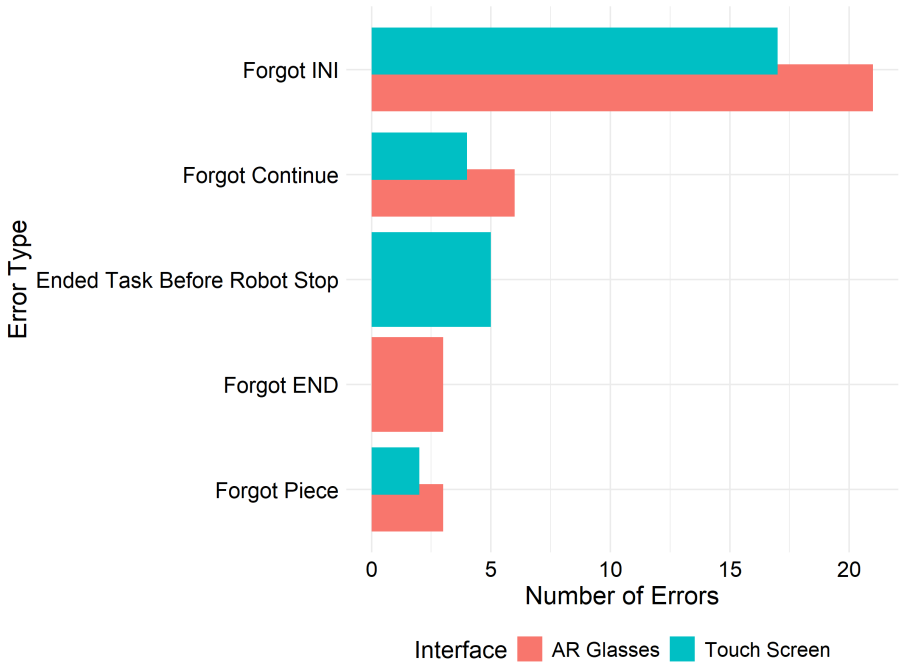


Figure 6.27: Frequency of errors categorized by type, highlighting the most common mistakes participants made during the tasks and providing insight into areas where interface or task design may need improvement.

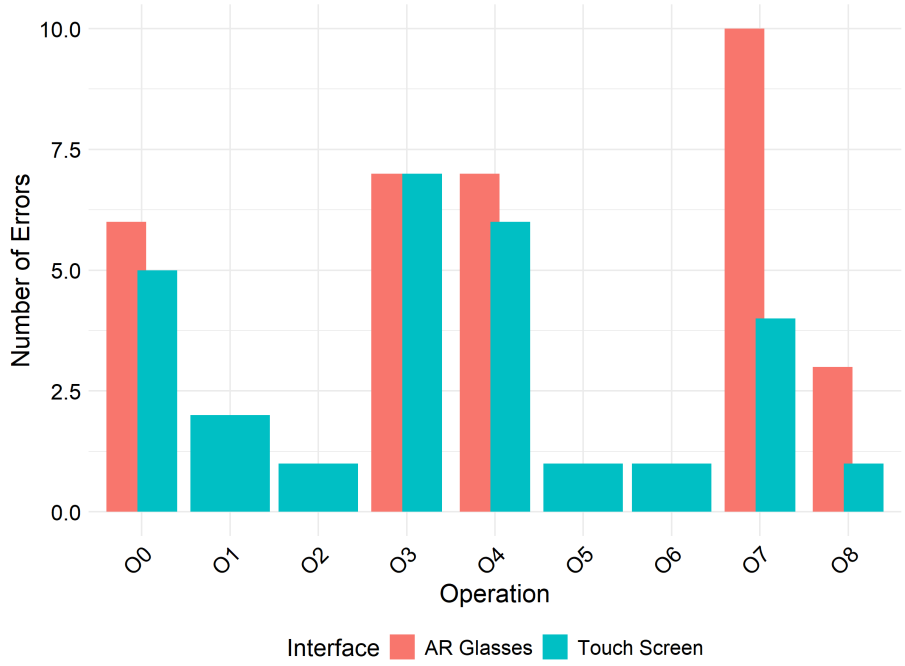


Figure 6.28: Number of errors occurring in each operation, showing which specific steps in the workflow were most prone to mistakes and indicating potential bottlenecks in task execution.

button input was required and the similar visual weight assigned to instructional content and confirmation controls were recurring observations across participants.

Execution Times

Total execution time was 49.3 s longer with the AR glasses (95% CI [26.5, 72.0], $p_{\text{Holm}} < 0.001$) once the session effect was accounted for, whereas total procedure time did not differ between interfaces ($p = 0.393$), as shown in Table 6.4 and Fig. 6.30. Prior experience did not affect either measure (Table B.10).

At the operation level (Table 6.6 and Fig. 6.31), the touch screen was faster in seven of the nine operations after Holm correction (O0, O1, O2, O4, O5, O6, and O8); O3 and O7, both human-exclusive, did not differ. The pattern differs by agent type. In the four robot-exclusive operations (O1, O2, O5, and O6), whose duration is dominated by robot motion that was identical in both conditions, AR added a small and consistent overhead of 1.5–4.6 s per operation, with no session effect. In the human-exclusive and collaborative operations, session effects were larger: the longest manual operation, O3, was 33.3 s faster in the second session ($p_{\text{Holm}} < 0.001$) with no significant interface effect, and O0, O4, and O8 showed nominal session effects that did not survive correction. Detailed statistics are given in Table B.18.

Practice and Order Effects

Because every participant performed the same disassembly twice, the sequence-specific means in Fig. 6.32 combine an interface effect with a practice effect. Participants who started with AR

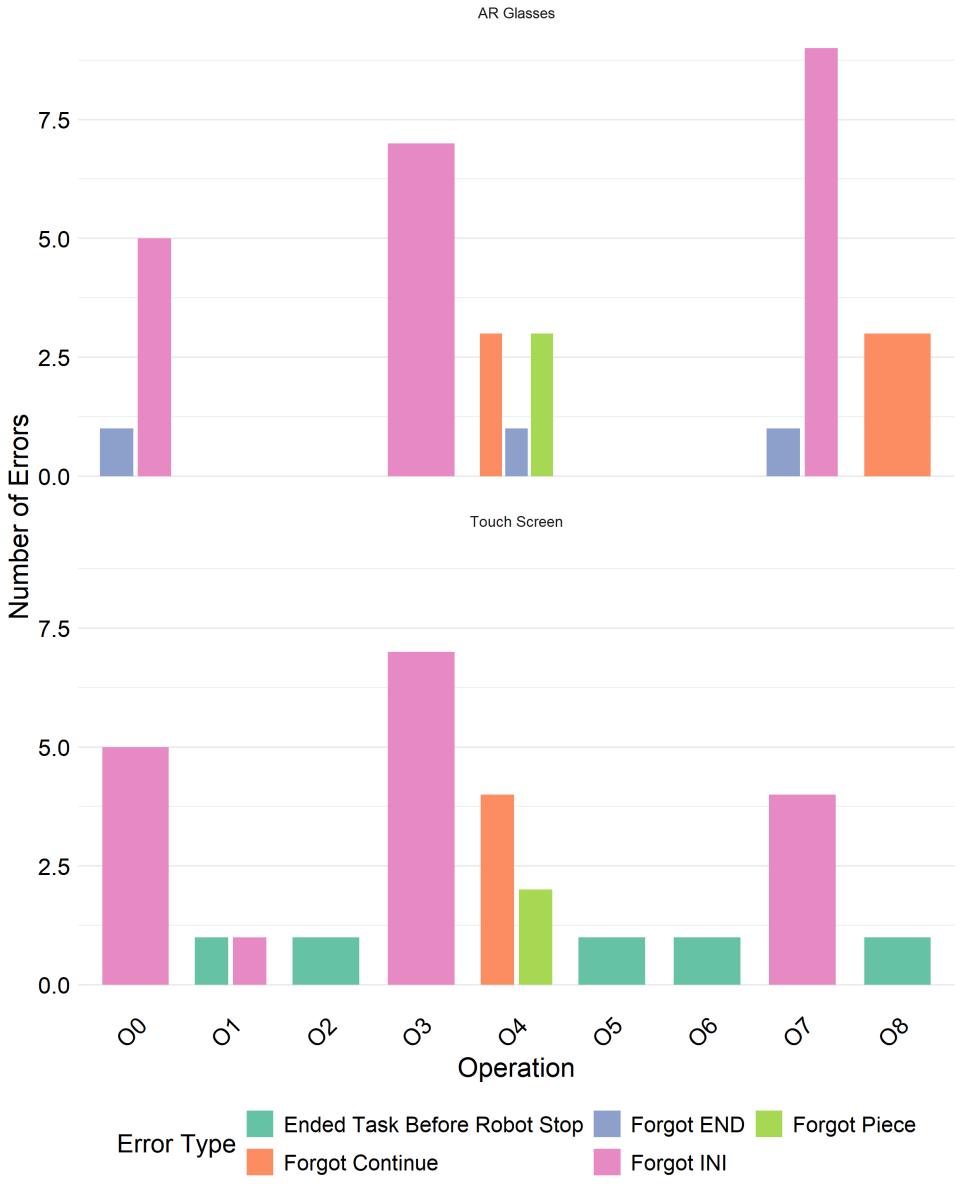


Figure 6.29: Combined analysis of errors by both operation and type, illustrating patterns in how different mistakes occur at specific workflow stages, useful for targeted interface or training improvements.

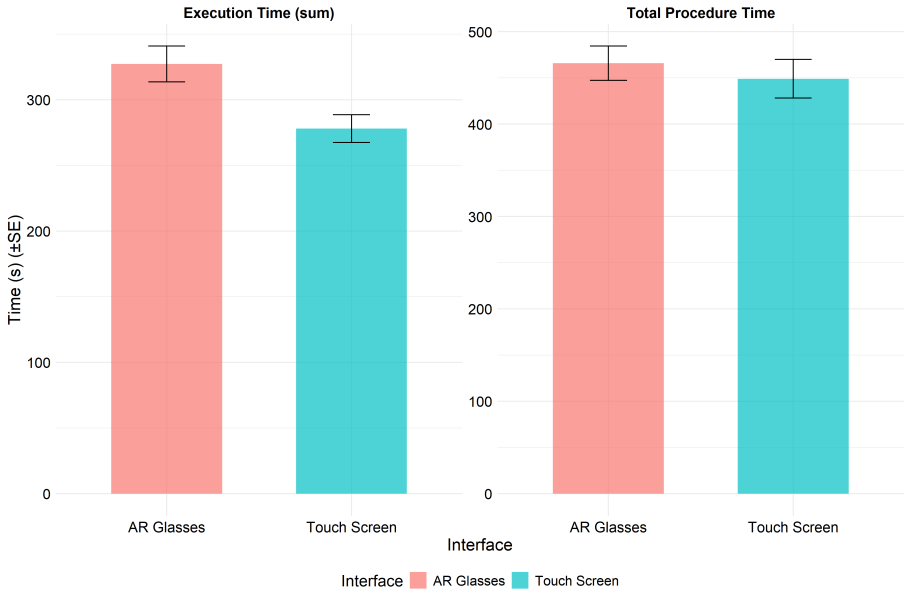


Figure 6.30: Mean execution and procedure times for all participants, compared across interfaces, highlighting how interface type impacts task duration and overall procedural efficiency.

Table 6.6: Execution time by operation (seconds): interface comparison and session effect. H: human-exclusive; R: robot-exclusive; C: collaborative. W: Wilcoxon signed-rank test; t: paired t-test ($n = 36$); Holm correction across the nine operations. Session effect from the crossover mixed model (second minus first session, unadjusted p); only the O3 session effect remains significant after Holm correction.

Operation	Agent	TS, M (SD)	AR, M (SD)	Test	p_{Holm}	d_z	Session Δ (p)
O0	H	9.6 (7.8)	17.3 (13.6)	W	0.001	0.60	−4.4 (0.040)
O1	R	21.9 (1.9)	24.8 (3.5)	W	<0.001	0.82	−0.4 (0.460)
O2	R	19.3 (1.9)	23.3 (5.3)	W	<0.001	0.74	0.9 (0.329)
O3	H	103.0 (53.3)	112.4 (51.8)	t	0.305	0.17	−33.3 (<0.001)
O4	C	37.0 (9.3)	48.3 (20.8)	W	0.017	0.49	−8.7 (0.021)
O5	R	21.7 (1.7)	26.3 (11.0)	W	<0.001	0.40	−0.7 (0.724)
O6	R	22.9 (1.9)	24.4 (2.7)	W	0.015	0.43	−0.1 (0.921)
O7	H	9.9 (4.2)	12.0 (5.3)	W	0.158	0.33	−0.7 (0.507)
O8	C	32.6 (6.5)	38.4 (13.3)	W	0.008	0.51	−3.9 (0.038)

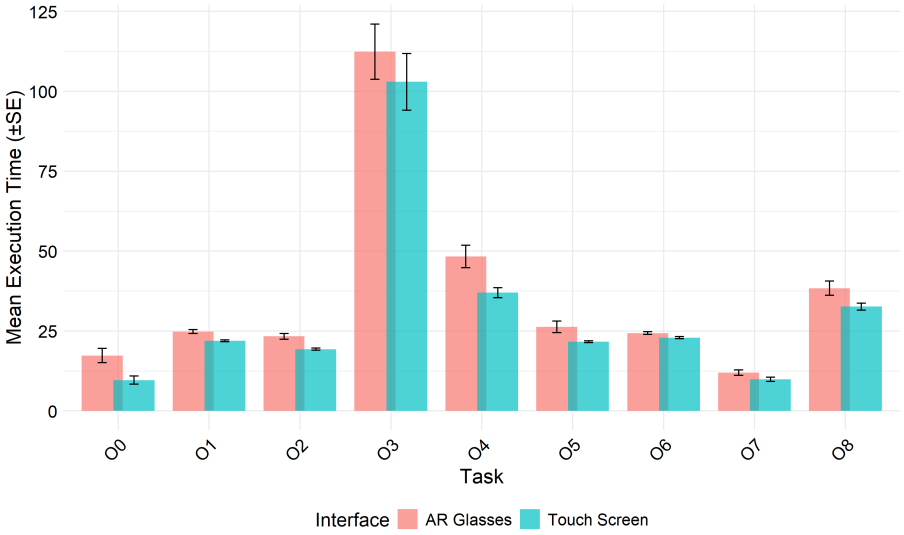


Figure 6.31: Mean execution times for individual operations compared across interfaces, showing consistent differences in task-specific durations between the AR glasses and Touch Screen conditions.

needed 354.7 s of execution time with AR and 254.2 s with the touch screen in their second session, a difference of about 100 s. Participants who started with the touch screen needed 301.8 s with the touch screen and 299.9 s with AR in their second session, a difference close to zero. The crossover model accounts for both patterns with two additive effects of similar size and opposite sign, an AR overhead of 49.3 s and a practice gain of 51.2 s: for participants who started with AR, both effects add up, whereas for those who started with the touch screen they cancel out. Procedure time followed the same logic, with a larger practice gain (119.5 s) and no interface effect.

The sequence term, which in this design captures differential carryover, was not significant for execution time ($p = 0.860$) or procedure time ($p = 0.234$). The data therefore do not indicate that practice with one interface transferred differently to the other, contrary to the self-reported impression that AR-first participants benefited more; given the limited power of this test (Section 6.7.2.1), a moderate asymmetry cannot, however, be excluded. For errors, the first session was the most error-prone in both sequences, and the difference between sessions within the AR-first group, significant before correction ($p = 0.044$), did not survive Holm correction across the two sequences ($p_{\text{Holm}} = 0.088$). Detailed sequence-specific analyses, including the operation-level results, are reported in Appendix B.6.

6.7.4.5 Exploratory Subgroup Analyses

Two exploratory patterns, reported in full in Appendix B.3, are noted for completeness. Participants with poor prior disassembly experience ($n = 7$) reported a higher NASA-TLX score including eye fatigue with AR than with the touch screen ($p = 0.035$, $d_z = 0.93$, uncorrected), whereas no difference was found in the other experience groups. Participants with above-average robot experience ($n = 20$) showed a tendency toward higher General Questions scores with AR

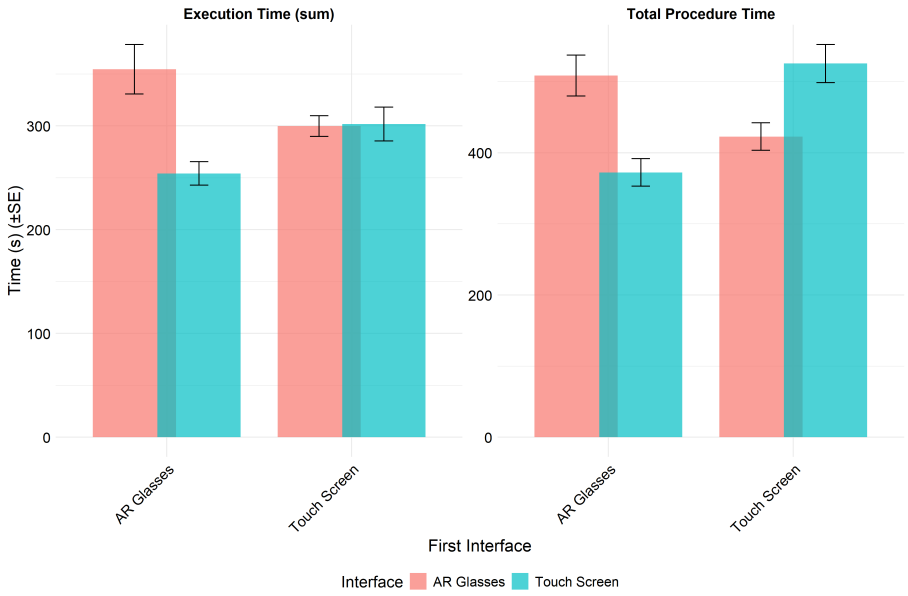


Figure 6.32: Mean execution and procedure times broken down by the first interface used, illustrating the effects of interface order on overall task duration and procedural efficiency.

($p = 0.092$), with the largest differences for the lower-experience groups observed in the feeling of being in charge of the task. Given the small group sizes and the absence of correction, these patterns are hypotheses for future work rather than findings.

6.8 Discussion

The results reveal a clear asymmetry between the two interface configurations. The AR configuration did not produce a global gain in usability or efficiency: the touch screen was rated as significantly more usable, produced shorter execution times, caused less eye fatigue, and was preferred by the majority of participants for day-to-day operation. Yet the AR configuration improved the dimension most directly tied to HRI quality, namely the operator's awareness of the robot's next action, and raised the overall perceived quality of working with the robot, with trends toward a stronger sense of collaboration and toward losing track of the robot less often. The following subsections interpret this asymmetry in terms of the framework of Section 6.1, relate it to prior work, and examine the role of practice.

6.8.1 What the AR Configuration Improved, and Why

The strongest and most robust HRI effect concerned awareness of the robot's next action, i.e., the projection level of SA (Level 3). This is the level that the trajectory preview directly supports: instead of inferring the robot's next motion from a pre-recorded video on a screen, the operator sees the planned path in the workspace, where it can be compared directly with the position of their own hands and of the parts. In the terms of the framework, the AR configuration externalizes

robot intent that the motion itself does not make legible [167], and it does so at the location where the information is used, removing the need to integrate two spatially separated sources [173], [174]. Participants' descriptions, seeing where the robot would move and stop, and feeling safer and more confident as a result, match this mechanism.

The weaker, non-significant effects on not losing track of the robot (Level 1) and on perceived collaboration are consistent with the same account, but indicate that projection benefits do not automatically propagate into a stronger sense of partnership. With a fixed task allocation and a robot that executed predefined actions, the scope for perceived collaboration was limited: the operator could anticipate the robot better, but could not negotiate with it. Anticipation is a prerequisite for fluent collaboration [172], and the observed trend suggests that its effect on perceived collaboration may become more pronounced in tasks with dynamic allocation or closer physical coupling, which remains to be tested. The absence of differences in work pace and attention allocation further indicates that AR changed what operators knew about the robot, rather than how the robot paced their work.

Two qualifications are necessary. First, awareness of the robot's next action also increased from the first to the second session, independently of the interface. Part of Level 3 SA therefore develops through familiarity with the robot's behavior, and the AR effect is best understood as additive to this learning rather than as a substitute for it. Second, because the AR configuration differed from the touch screen in several respects at once (Table 6.2), the attribution of the awareness gain to the trajectory preview, rather than to color-coded ownership or to in-situ instructions in general, rests on the item content and on the interviews, not on an experimental separation of features.

6.8.2 Why the Same Configuration Increased Fatigue and Reduced Usability

The benefits and the costs of the AR configuration arise at different layers of the interaction, which explains how both can occur simultaneously. The benefits are informational: they concern what the operator knows about the robot and the task, which is what the Working with the Robot items measure. The costs are perceptual and motor: they concern how the operator sees and acts on the interface, which is what the eye-fatigue item and the SUS capture. An interface can therefore improve robot awareness while being harder to use, because the two properties are produced by different components of the same configuration.

On the perceptual side, eye fatigue was the largest and most consistent cost ($d_z = 0.81$) and was the only NASA-TLX item that differed after correction; the raw NASA-TLX did not differ, indicating that the general cognitive demand of the two configurations was comparable. Optical see-through HMDs present holograms at a fixed focal distance, and the vergence-accommodation conflict that arises when the eyes alternate between holograms and nearby physical parts is a known source of visual fatigue [175]. In a disassembly task, this alternation is intrinsic: the operator must look at the hologram and then at the physical part it refers to. Participants' reports of opaque, highly saturated holograms that occluded the parts they had to manipulate point to a

second, design-related contributor. Notably, this fatigue emerged after sessions lasting less than ten minutes, which raises concerns for shift-length use that the present study could not assess.

On the motor side, the SUS items that differed after correction concerned the need for technical support and the speed of learning, i.e., learnability rather than the integration or consistency of the functions. HL2 interaction (pinch gestures, close-range virtual buttons, and the virtual keyboard) was widely described as inconsistent and lacking confirmation. Without tactile feedback, operators were uncertain whether inputs had been registered, which likely contributed to frustration and repeated inputs. The operation-level execution times support this interpretation: in the robot-exclusive operations, whose duration is determined by robot motion identical in both conditions, AR added a constant overhead of a few seconds without any practice effect, which is best explained by the time needed to issue and verify confirmations with mid-air input. Conversely, the familiarity of touch interaction from everyday devices and the tactile confirmation of the screen likely explain much of the touch screen's usability advantage. These explanations were not tested independently and should be read as likely rather than established causes.

Both configurations also shared design limitations that affected performance independently of the device: dense information, weak visual hierarchy, and confirmation controls that did not signal when they were required. This is reflected in the dominant error type, forgetting to press a required button, which accounted for 51 of the 61 errors and occurred at similar rates with both interfaces. Together with the robot-operation overhead, this indicates that differences in time and errors were driven mainly by the interaction flow and its input mechanisms, not by differences in task comprehension, in line with expectation E3.

6.8.3 Comparison with Previous Studies

The pattern observed here is consistent with, and helps reconcile, several findings in the literature. As in Chu and Liu [19], where a preview of the gripper's future position reduced gaze time toward the robot, the trajectory preview mainly supported anticipation of robot motion. As in Lunding, Lystbæk, Feuchtner, *et al.* [26], from which the questionnaire items were taken, participants emphasized the value of task-ownership visualization, NASA-TLX workload did not differ between conditions, and eye fatigue was higher with AR; the present study replicates this pattern in a disassembly task, with a larger crossover sample and with a baseline that provides the full functionality of an industrial HMI. The preference for a projector over an HMD reported by Hietanen, Pieters, Lanz, *et al.* [55], and the influence of input technique on workload and usability reported by Tadeja, Zhou, Capponi, *et al.* [138] and Sanna, Manuri, Fiorenza, *et al.* [139], mirror the usability costs observed here.

Table 6.7 summarizes how the present study differs from the closest AR-based HRC studies. The usability results also fit this comparison. The SUS score of the AR configuration (74.7) lies at the lower end of the values reported for HMD-based HRC interfaces in Table 3.3 (76–82) [138], [139], [142], [143], whereas the touch screen obtained 83.3. Those studies did not include an industrial HMI as the reference condition. The absolute AR score is therefore of a similar order

to previous reports, and the usability difference observed here is more plausibly explained by the strength of the baseline than by a particularly weak AR implementation.

Table 6.7: Comparison of the present user study with the closest AR-based HRC studies reviewed in Section 3.2.

Study	Reference condition	Task allocation	Device	Measures
[55]	Non-collaborative baseline; projector	–	HL1; projector	Execution time, robot idle time, custom UX questionnaire
[54]	Screen-based feedback	AOG; A*	HL1	Applied force, custom usability questionnaire
[130]	Manual baseline	AOG; A*	HL1	Custom usability questionnaire
[25]	Manual baseline	BT; MILP	HL2	Execution time, NASA-TLX, custom questionnaire
[26]	Tablet	Operator-mediated	HL2	Execution time, NASA-TLX, custom questionnaire, interview
[19]	Other AR interface; haptic feedback	–	HL2; SenseGlove	Execution time, hand proximity, gaze time, SUS, safety questionnaire
[138]	Other AR input method	–	HL2	Execution time, SUS, NASA-TLX, SSQ
This study	Functionally matched industrial touch screen HMI; crossover design	MILP scheduler (fixed allocation in the study)	HL2; touch screen	Execution and procedure time, errors, SUS, NASA-TLX with eye fatigue, custom questionnaire (robot awareness and collaboration), interview; interface and practice effects separated

The results diverge from studies in which AR reduced workload or execution time [25], [130], [136]. A likely reason is the reference condition: when AR is compared with a manual baseline, or with a screen that provides feedback only, AR contributes both information and interaction support, whereas a competent, familiar HMI already provides the instructional content and a low-cost input channel. Against such a baseline, the advantage of AR narrows to what the screen cannot provide, namely spatially anchored robot intent, while its interaction costs remain. This suggests that the merit of AR in industrial HRC lies specifically in robot-related SA rather than in general efficiency or workload reduction.

The present work also complements recent XR research on assembly and disassembly. A state-of-the-art review of AR guidance for assembly and maintenance/repair [201] analyzes the hardware and software elements of such systems and identifies tracking and registration as a central challenge; the present study adds user-level evidence on another practical barrier, the perceptual and input costs of HMD-based guidance in a collaborative setting. Eswaran and Raju Bahubalendruni [183] addressed the effort of creating AR content with a framework that renders the virtual scenes and model targets used to create AR disassembly guidance in Unity, and Cantarelli, Francia, Santi, *et al.* [202] developed a mixed reality laboratory for assembly and disassembly training aimed at shorter training times and fewer errors. These studies focus on how XR content is produced and on operator training, and do not involve a collaborating robot or a comparison with an

industrial HMI. The present study contributes what they do not address: a controlled user evaluation of AR in a disassembly task shared with a robot, against a functionally matched touch screen baseline, with interface and practice effects separated. Conversely, their frameworks point to how the main scalability limitation identified here, the authoring cost of AR content per product, could be reduced.

6.8.4 Practice Effects and Implications for Training

Practice effects were as large as interface effects on execution time and larger than any interface effect on procedure time, and they also reduced workload, particularly mental demand. Any comparison in which the same task is repeated must therefore be interpreted with the order of conditions in mind: a naive comparison within the AR-first group would have overstated the AR execution-time cost by a factor of two, and a comparison within the touch screen-first group would have suggested no cost at all. The crossover analysis shows that both groups are consistent with the same additive effects.

This has direct consequences for the interpretation of AR as a training tool. Half of the participants, and all experts, preferred AR for training and onboarding, and several believed that learning with AR first transferred better to the touch screen. The performance data do not support a differential transfer: the practice gain was similar regardless of which interface was used first, and the first session was the most error-prone in both sequences, most markedly when it was performed with AR. XR environments have been proposed specifically to shorten operator training in assembly and disassembly [202], and the preference for AR in onboarding observed here is consistent with that motivation. It is, however, best interpreted as a perceived benefit of in-situ instructions and robot-intent cues during the formation of a mental model of the task, which is plausible given the awareness results, but which remains to be demonstrated with appropriate designs, i.e., with novice operators, different tasks for training and testing, and retention measures.

6.8.5 Implications for Interface Design in Industrial Human-Robot Collaboration

The most practically significant implication of these results is not that AR is better or worse than a touch screen, but that the information an interface provides and the way the operator interacts with it are separable design dimensions with different effects. The AR configuration provided spatial robot-intent information that improved awareness of the robot's actions, but its interaction methods imposed costs that outweighed those benefits in terms of overall usability and task efficiency. The touch screen provided an efficient and familiar interaction, but limited the operator's ability to anticipate robot behavior in the shared workspace.

These findings point toward hybrid or multimodal interface designs, combining the interaction efficiency of touch-based or physical input with the spatial visualization strengths of AR. The participant and expert recommendations gathered during the study provide concrete design guidance for such an approach. Across both interfaces, the most consistent recommendations concerned

Table 6.8: Design guidelines for operator interfaces in industrial HRC derived from the user study, grouped by interface type, and the general principles that motivate them.

Scope	**Guideline**
AR interface	• Avoid mid-air gestures for frequent confirmations; use touch or physical buttons instead. • Reduce hologram opacity, or use wireframe overlays, during manual manipulation; allow color and saturation to be adjusted. • Prioritize the communication of robot intent (planned trajectory and task ownership). • Provide a virtual safety fence that alerts the operator when entering the robot's workspace during motion. • Keep critical controls in the central field of view.
Touch screen	• Enlarge instructional videos and offer an optional full-screen mode. • Integrate text labels into the video to reduce attention shifts. • Add graphical task-ownership indicators, analogous to the AR color coding. • Give critical controls a more prominent position in the visual hierarchy.
Both	• Replace the multi-step confirmation sequence with a single state-changing control (START → NEXT → END). • Add a progress bar that communicates task completion status. • Detect task completion automatically where possible. • Combine visual, haptic, and auditory feedback. • Reduce information density and establish a clear visual hierarchy.
General principles	• Spatial information about the robot (trajectory, intent, and task ownership) is a design requirement, not an optional enhancement. • Interaction matters as much as visual content: what is displayed and how the operator interacts with it must be designed together; input feedback, gesture consistency, and field of view determine adoption. • AR and touch screens are complementary: AR for training and onboarding, touch screens for routine operation, and hybrid designs (touch input with AR visualization) to keep the HRI gains without the interaction cost.

workflow transparency, such as replacing the multi-step button confirmation sequence with a single dynamic state-changing control (e.g., START → NEXT → END), introducing a progress bar to communicate task completion status, and implementing automatic task completion detection to eliminate the need for manual confirmation presses. Multimodal feedback integrating visual, haptic, and auditory signals was also advocated, as was multi-language support and greater task parallelism to allow the operator to begin subsequent actions while the robot completes its current task. For the AR interface specifically, participants and experts consistently recommended wireframe or semi-transparent holographic overlays to reduce visual occlusion during manual manipulation, along with customizable color and saturation settings to accommodate different task contexts and accessibility needs. A virtual safety fence alerting the operator upon entering the robot's workspace during motion was proposed as a means of supporting robot awareness without requiring continuous visual monitoring. For the touch screen, recommendations centered on improving visual hierarchy by placing critical controls more prominently, enlarging instructional

videos with an optional full-screen mode, and integrating text labels directly into the video feed to reduce attention-shifting. Several experts also proposed extending the touch screen with graphical task-ownership indicators analogous to the AR color-coding scheme, further narrowing the HRI information gap between the two modalities. These recommendations, together with the general principles that emerge from the study, are summarized in Table 6.8.

Regarding scalability, the separation between product-independent logic and product-specific content (Section 6.6) means that both configurations extend to other products and disassembly sequences without changes to the scheduler or the confirmation flow, but the AR configuration carries a higher authoring cost per product. For deployments with many product variants, this cost, together with the interaction costs discussed above, argues for reserving AR for the information that only it can provide well, such as robot intent and task ownership in the shared workspace.

What the present study demonstrates experimentally is limited to one collaborative workstation, one product, a fixed task allocation, and short sessions. The mechanisms identified, i.e., the need to anticipate robot motion in a shared space and the cost of mid-air input and HMD viewing, are also present in collaborative assembly, inspection, and material-handling tasks, so the design implications above are plausible hypotheses for those scenarios. Whether the balance between benefits and costs holds there depends on factors that were not varied here, such as task duration, operator mobility, lighting, protective equipment, and the frequency of robot motion near the operator.

6.9 Limitations and Future Research

The results should be interpreted in light of several limitations, each of which suggests a direction for future research. The first concerns the design of the comparison itself. The two conditions differed in visualization content, display, input, and feedback (Table 6.2), so the study estimates the effect of complete configurations, and the contribution of individual features, such as the trajectory preview, color-coded ownership, or mid-air input, cannot be separated. Isolating them requires feature-level designs, for instance adding a trajectory preview or ownership indicators to a screen-based interface, or pairing AR visualization with touch or physical input, which would also allow the hybrid interfaces suggested by the results to be evaluated.

The generality of the findings is further limited by the sample. Participants were recruited from engineering universities and research institutes, and were predominantly male, highly educated, engineering-oriented, and experienced with robots. Industrial operators may differ in their familiarity with touch devices, AR, and robots, in their tolerance for interface changes, and in their motivation, and the direction of these differences is uncertain: less robot experience may increase the value of robot-intent cues, whereas less digital experience may increase the cost of mid-air interaction and the time needed to learn it. The size of the sample also bounds what the study could detect. With 36 participants, it was adequately powered for medium within-subject effects only, so the trends observed for perceived collaboration and for losing track of the robot ($d_z \approx 0.34$) may reflect true effects that remained undetected, and the test for differential carryover had low power.

Replication with larger samples of industrial operators is therefore required before the findings can be generalized to production staff.

The way in which participants were exposed to the interfaces constrains the interpretation of the time and comfort results. Each participant performed the same disassembly twice; although the crossover model separates additive practice effects from interface effects, it cannot fully exclude differential carryover, and the second session was, in part, a test of memory for the procedure. Moreover, each AR session lasted less than ten minutes of task time with a single device (HL2), whose field of view, optics, and input methods shaped the costs observed, so the eye-fatigue and usability results concern short exposures and may not hold for longer sessions or other devices. Future studies should use different but equivalent task configurations in each session, include training and retention phases, and follow operators over longer periods of AR use to establish how much of the observed interaction cost is due to novelty.

The measurement instruments also leave room for improvement. The custom questionnaire used single items adapted from prior work, the Working with the Robot and AR Visualization blocks showed low internal consistency, and the item-level results, though corrected for multiplicity, rely on single ratings; in addition, the AR Visualization block had no touch screen counterpart. Validated instruments for SA, fluency, and trust, together with objective indicators such as gaze behavior or response times to robot events, would strengthen the measurement of HRI quality. On the qualitative side, the interviews were short (about ten minutes) and were analyzed by a single researcher, without a predefined coding framework or a second coder, so the reported topics and their prevalence are indicative, and some less frequent topics may have been missed or interpreted differently by another analyst.

Finally, the experimental setting favored internal validity over ecological completeness. The experiment used a single collaborative workstation, a simplified 3D-printed alternator, a fixed task allocation, and a laboratory setting with simulated MPC arrivals. Extending the comparison to multi-station MMS scenarios with dynamic task allocation, to real products with heavier and more varied components, to partial or selective disassembly sequences generated by DSP methods, and to realistic production environments is a natural next step. In such settings, a co-design approach involving shop-floor operators in the design of icons, colors, terminology, and workflow logic would help ensure that interface solutions are validated against the operational culture and practical constraints of real manufacturing environments.

6.10 Operator Interfaces Conclusions

This chapter compared an AR interface and a conventional touch screen HMI, as complete interface configurations, in a collaborative disassembly task within an MMS, with the primary objective of investigating how the interface shapes HRI quality. The AR configuration did not achieve global superiority in usability or efficiency, but it improved operators' awareness of the robot's next action and the overall perceived quality of working with the robot, with trends toward a stronger

sense of collaboration. These advantages are consistent with the AR configuration's in-situ robot-intent cues, in particular the trajectory preview and color-coded task ownership, as described by participants, although the design of the study does not allow the contribution of each feature to be isolated.

These HRI advantages came with a measurable interaction cost. The AR configuration caused significantly greater eye fatigue, received significantly lower usability ratings, particularly regarding learnability and the need for support, and resulted in longer execution times, partly due to a constant confirmation overhead in robot-exclusive operations. General workload did not differ between interfaces. The touch screen was preferred by the majority of participants for day-to-day operation.

A result that cuts across both conditions is the size of the practice effects: repeating the task reduced execution time by as much as the interface increased it, and reduced procedure time and workload, with no evidence that practice transferred differently depending on the interface used first. Errors were dominated by missed confirmation presses in both conditions, indicating that the interaction flow, rather than task comprehension, determined performance and error patterns.

Three practical implications follow for interfaces in industrial HRC. First, spatial information about robot behavior, such as trajectory, intent, and task ownership, improves the operator's awareness of the robot and should be treated as a design requirement rather than an optional enhancement. Second, the input and feedback mechanisms through which the operator interacts with that information matter as much as the information itself, and determine usability and efficiency. Third, the complementary strengths of the two configurations point toward hybrid designs that use AR for robot-intent visualization and touch-based or physical input for routine interaction. Whether these implications extend to assembly, inspection, logistics, and material-handling scenarios, where operator-robot coordination is equally critical, remains to be established with industrial operators, longer exposures, and feature-level comparisons.

Chapter 7

Conclusions and Future Work

This thesis addressed two interrelated challenges in the design of HRC systems for flexible MMSs: how to coordinate heterogeneous HRTs, mobile robots, and battery-constrained resources through optimization, and how to support the human operator within that coordinated system through interfaces that improve awareness of the robot and the quality of interaction. This chapter answers the research questions posed in Section 1.3, summarizes the contributions, and outlines directions for future work.

7.1 Answers to the Research Questions

7.1.1 RQ1: How effectively do integrated MILP formulations for task scheduling, robot relocation, and battery management support efficient and coordinated operation in MMSs?

Within the scale tested, effectively. On the deployed Prolynk MMS, the three sequentially solved modules produced feasible and conflict-free plans for a nine-operation disassembly with four agents and three MPCs: a makespan of 343 s and a flowtime of 1683 s in the first cycle, the selection of MPC 3 for a 32 s battery swap (28% to 100%), and relocations that freed the required workstations. Re-solving after each operation completion allowed the plan to follow the physical state of the system, as when MPC 3 was redirected after operation 0, saving 3 s. The limits of the method define its range of application: the Task Scheduling module required 34.74 s per cycle, and its solve time grows with the number of jobs, agents, and workstations; the sequential decomposition does not guarantee global optimality; and reactions are bounded by the indexing points of the mobile robots. The framework is therefore effective for small- to medium-scale deployments under the tested conditions, while larger and more dynamic systems require the extensions discussed in Section 7.3.

7.1.2 RQ2: To what extent does the AR interface improve user experience and performance compared to a conventional HMI?

Selectively. Compared with a functionally matched touch screen HMI, the AR interface improved the perceived quality of working with the robot (Working with the Robot index, +0.25 on a 1–5 scale) and, in particular, the operator's awareness of the robot's next action, with trends toward a stronger sense of collaboration. It did not improve usability or efficiency: the touch screen obtained a significantly higher SUS score (83.3 versus 74.7), caused less eye fatigue, and led to shorter execution times, whereas general workload, total procedure time, and the number of errors did not differ. Most participants preferred the touch screen for routine operation, and half preferred AR for training and onboarding.

7.1.3 RQ3: Which factors contribute to the improvement or deterioration of user experience and performance?

The improvements are consistent with the in-situ robot-intent cues of the AR interface, namely the trajectory preview, the color-coded task ownership, and instructions anchored to the physical parts, although the study design does not isolate the contribution of each feature. The deteriorations are associated with how the operator sees and acts on the interface: mid-air input without haptic feedback, the limited field of view and HMD viewing, opaque holograms, and device weight. Interaction design affected performance more than visualization modality: 51 of the 61 errors were missed confirmation presses in both interfaces, and AR added a constant overhead even in robot-only operations. Practice reduced execution time by as much as the AR interface increased it, with no evidence that practice transferred differently depending on the interface used first.

7.2 Contributions

7.2.1 C1 – Knowledge synthesis on HRT scheduling and AR-supported HRC

A structured review showing that task allocation, mobile robot logistics, battery management, and collaborative execution modes are treated in isolation, and that AR is rarely compared with an equivalent industrial HMI using HRI-specific measures (Chapter 3).

7.2.2 C2 – Benchmarking of fiducial marker tracking on the HL2

A comparison of Vuforia, ARToolKit, ArUco, and QR Code tracking against OptiTrack ground truth at 25 positions, establishing accuracy and precision trade-offs and supporting the selection of QR Code tracking for the close-range workstation (Section 4.4; [17]).

7.2.3 C3 – MILP-based robotic coordination framework for HRTs in MMSs

Three sequentially solved MILP modules for battery swapping, task scheduling, and mobile robot relocation, combining FJSSP sequencing, multi-mode execution by humans, robots, or both, mobile robot logistics, and energy constraints, and validated on a deployed industrial MMS (Chapter 5; [21]).

7.2.4 C4 – Comparative evaluation of immersive and non-immersive interfaces

A counterbalanced within-subject study with 36 participants comparing an AR interface with a functionally matched industrial touch screen HMI, analyzed with crossover mixed models and multiplicity control, which characterizes the trade-off between HRI quality and usability and grounds the design guidelines for AR, touch screen, and hybrid interfaces in Table 6.8 (Chapter 6; [22]).

7.3 Future Work

The directions for future work fall into three categories: direct continuation of the main contributions, methodological improvements to the user study, and extension of the system to more realistic conditions.

7.3.1 Direct Continuation of the Main Contributions

The most immediately actionable direction for the coordination framework is the investigation of tighter coupling between the Battery Swap and Task Scheduling modules. The current sequential decomposition can produce suboptimal schedules when a robot assigned to a closer ABC is subsequently needed at a far workstation; a partially integrated or iterative formulation could reduce this gap without sacrificing the tractability benefits of separation. To address scalability, CP and metaheuristic approaches such as GA, simulated annealing, and PSO should be investigated as alternatives to the Branch-and-Cut solver for larger instances where exact methods exceed the scheduling window. Artificial intelligence techniques offer more targeted solutions. RL is a natural candidate for online rescheduling, adapting dispatch decisions in response to dynamic events such as task delays, robot failures, or agent unavailability without restarting a full optimization cycle; neural networks could predict operation execution times, workstation congestion, or battery consumption patterns to improve the accuracy of the scheduling model's inputs; and learned heuristics could serve as warm-start strategies for the MILP solver, narrowing the search space before branch-and-cut exploration begins. On the hardware side, equipping the platform with real-time robot localization and mid-transit rerouting capability would remove the most significant constraint on the framework's responsiveness to dynamic disturbances.

For the interfaces, the user study compared complete configurations, so it shows which combination of features worked better but not which feature produced each effect. The next step is therefore a set of feature-level comparisons: adding the trajectory preview and color-coded task

ownership to a screen-based interface, and pairing AR visualization with touch or physical input instead of mid-air gestures. Such comparisons would establish whether the gain in robot awareness can be kept while removing the interaction costs, which is the premise of the hybrid designs suggested by the results, and they would test the design guidelines of Table 6.8 directly. Within the current HMD platform, the refinements proposed by participants and experts can be implemented and evaluated in the same way: lower hologram opacity or wireframe overlays during manual manipulation, a single state-changing confirmation control with clearer feedback, stronger visual hierarchy for critical controls, and consistent positioning of panels and keyboards. A co-design process with shop-floor operators would ground these choices in the conventions and constraints of real production environments.

The scalability of the AR interface to many products depends mainly on content authoring, since its animations are currently built into the application. Loading animation assets at runtime and generating them from CAD data and the disassembly sequence, as in frameworks that render the virtual scenes and model targets for AR disassembly guidance [183], would allow new products and variants to be added without rebuilding the application; the effort required, and the quality of automatically generated content, remain to be evaluated.

7.3.2 Methodological Improvements to the User Study

Several aspects of the experimental design limit the strength of the conclusions and should be addressed in follow-up studies. First, participants repeated the same fixed disassembly sequence in both sessions; the crossover analysis separates additive practice effects from interface effects, but different yet equivalent task configurations in each session would remove the memory component of the second session. Training and retention phases, and longer periods of AR use, would show how much of the observed interaction cost is due to novelty. Second, the sample was drawn from engineering universities and research institutes and was experienced with robots; replication with larger samples of industrial operators is required before the results can be generalized to production staff, and a larger sample would also allow the small effects observed for perceived collaboration and for losing track of the robot to be tested with adequate power. Third, HRI quality was measured mainly with single questionnaire items, some of which formed blocks with low internal consistency. Validated instruments for situation awareness, fluency, and trust, combined with objective indicators such as gaze behavior and response times to robot events, would provide a stronger measurement, and interviews analyzed with a predefined coding framework and a second coder would make the qualitative findings more reliable. Finally, the eye-fatigue and usability results concern sessions of less than ten minutes with a single device; longer sessions and other HMDs are needed to establish how these costs evolve with exposure and hardware.

Behavioral instrumentation of the instructional content would also strengthen future studies. Recording whether participants watch the full instructional video or animation, or skip segments, and comparing execution times with a minimum time derived from content duration and average execution time, would distinguish efficiency gains from skipped instructions and explain part of the variation between participants that is currently attributed to the interface or to practice. The

same data could support adaptive interfaces that adjust the amount of instruction to the operator's engagement, offering more concise content to operators who consistently skip it and additional guidance to those who replay it.

7.3.3 Extension of the System

The most significant extension is the evaluation of the complete system, scheduler and interfaces together, in a multi-station MMS with dynamic task allocation and physical MPC circulation. The user study deliberately used a single collaborative workstation, a fixed task allocation, and simulated task arrivals, which favored internal validity over ecological completeness. In a multi-station scenario, the allocation of each operation would change at runtime, and the interface would have to communicate not only robot motion but also changes in task ownership decided by the scheduler; how well each interface supports the operator under these conditions is an open question. Extending the scheduling framework to coordinate several collaborative workstations simultaneously is a prerequisite for this evaluation. Further extensions include real products with heavier and more varied components, partial or selective disassembly sequences generated by DSP methods instead of a fixed complete sequence, and realistic production environments, where task duration, operator mobility, lighting, and protective equipment may change the balance between the benefits and costs observed in this thesis.

References

[1] M. Hermann, T. Pentek, and B. Otto, "Design principles for industrie 4.0 scenarios," in *2016 49th Hawaii International Conference on System Sciences (HICSS)*, 2016, pp. 3928–3937. DOI: 10.1109/HICSS.2016.488.

[2] V. Alcácer and V. Cruz-Machado, "Scanning the industry 4.0: A literature review on technologies for manufacturing systems," *Engineering Science and Technology, an International Journal*, vol. 22, no. 3, pp. 899–919, 2019, ISSN: 2215-0986. DOI: 10.1016/j.jestch.2019.01.006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2215098618317750.

[3] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and industry 5.0—inception, conception and perception," *Journal of Manufacturing Systems*, vol. 61, pp. 530–535, 2021. DOI: 10.1016/j.jmsy.2021.10.006.

[4] J. Leng, W. Sha, B. Wang, *et al.*, "Industry 5.0: Prospect and retrospect," *Journal of Manufacturing Systems*, vol. 65, pp. 279–295, 2022, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2022.09.017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612522001662.

[5] M. C. May, S. Schmidt, A. Kuhnle, N. Stricker, and G. Lanza, "Product generation module: Automated production planning for optimized workload and increased efficiency in matrix production systems," *Procedia CIRP*, vol. 96, pp. 45–50, 2021. DOI: 10.1016/j.procir.2021.01.050.

[6] A. Göppert, E. Schukat, P. Burggräf, and R. H. Schmitt, "Agile hybrid assembly systems: Bridging the gap between line and matrix configurations," in *Advances in Automotive Production Technology – Theory and Application*, P. Weißgraeber, F. Heieck, and C. Ackermann, Eds., Springer, Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 3–11, ISBN: 978-3-662-62962-8. DOI: 10.1007/978-3-662-62962-8_1.

[7] C. Ferreira, G. Figueira, and P. Amorim, "Scheduling human-robot teams in collaborative working cells," *International Journal of Production Economics*, vol. 235, p. 108 094, 2021, ISSN: 0925-5273. DOI: 10.1016/j.ijpe.2021.108094.

[8] M. M. Rahman, F. Khatun, I. Jahan, R. Devnath, and M. A.-A. Bhuiyan, "Cobotics: The evolving roles and prospects of next-generation collaborative robots in industry 5.0," *Journal of Robotics*, vol. 2024, no. 1, p. 2 918 089, 2024. DOI: 10.1155/2024/2918089.

[9] G. d. M. Costa, M. R. Petry, and A. P. Moreira, "Augmented reality for human–robot collaboration and cooperation in industrial applications: A systematic literature review," *Sensors*, vol. 22, no. 7, 2022, ISSN: 1424-8220. DOI: 10.3390/s22072725.

[10] D. Romero, P. Bernus, O. Noran, J. Stahre, and Å. Fast-Berglund, "The operator 4.0: Human cyber-physical systems & adaptive automation towards human-automation symbiosis work systems," in *IFIP international conference on advances in production management systems*, Springer, 2016, pp. 677–686. DOI: 10.1007/978-3-319-51133-7_80.

[11] D. Romero, J. Stahre, T. Wuest, *et al.*, "Towards an operator 4.0 typology: A human-centric perspective on the fourth industrial revolution technologies," in *proceedings of the international conference on computers and industrial engineering (CIE46), Tianjin, China*, 2016, pp. 29–31.

[12] I. Zolotová, P. Papcun, E. Kajáti, M. Miškuf, and J. Mocnej, "Smart and cognitive solutions for operator 4.0: Laboratory h-cpps case studies," *Computers & Industrial Engineering*, vol. 139, p. 105 471, 2020, ISSN: 0360-8352. DOI: 10.1016/j.cie.2018.10.032. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835218305126.

[13] D. Mourtzis and J. Angelopoulos, "Development of an extended reality-based collaborative platform for engineering education: Operator 5.0," *Electronics*, vol. 12, no. 17, p. 3663, 2023. DOI: 10.3390/electronics12173663.

[14] B. Gladysz, T.-a. Tran, D. Romero, T. van Erp, J. Abonyi, and T. Ruppert, "Current development on the operator 4.0 and transition towards the operator 5.0: A systematic literature review in light of industry 5.0," *Journal of Manufacturing Systems*, vol. 70, pp. 160–185, 2023, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2023.07.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612523001401.

[15] A. Blaga, C. Militaru, A.-D. Mezei, and L. Tamas, "Augmented reality integration into mes for connected workers," *Robotics and Computer-Integrated Manufacturing*, vol. 68, p. 102 057, 2021.

[16] E. Azimi, L. Qian, N. Navab, and P. Kazanzides, "Alignment of the virtual scene to the tracking space of a mixed reality head-mounted display," *arXiv preprint arXiv:1703.05834*, 2017. DOI: 10.48550/arXiv.1703.05834.

[17] G. M. Costa, M. R. Petry, J. G. Martins, and A. P. G. Moreira, "Assessment of multiple fiducial marker trackers on hololens 2," *IEEE Access*, vol. 12, pp. 14 211–14 226, 2024. DOI: 10.1109/ACCESS.2024.3356722.

[18] F. De Pace, F. Manuri, A. Sanna, and C. Fornaro, "A systematic review of augmented reality interfaces for collaborative industrial robots," *Computers & Industrial Engineering*, vol. 149, p. 106 806, 2020, ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106806. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835220305118.

[19] C.-H. Chu and Y.-L. Liu, "Augmented reality user interface design and experimental evaluation for human-robot collaborative assembly," *Journal of Manufacturing Systems*, vol. 68, pp. 313–324, 2023. DOI: 10.1016/j.jmsy.2023.04.007.

[20] G. V. Tchane Djogdom, R. Meziane, and M. J.-D. Otis, "Robust dynamic robot scheduling for collaborating with humans in manufacturing operations," *Robotics and Computer-Integrated Manufacturing*, vol. 88, p. 102 734, 2024, ISSN: 0736-5845. DOI: 10.1016/j.rcim.2024.102734.

[21] G. d. M. Costa, G. Figueira, A. P. Moreira, and M. R. Petry, "A robotic coordination framework for human–robot teams in matrix manufacturing," *Applied Sciences*, vol. 16, no. 14, 2026, ISSN: 2076-3417. DOI: 10.3390/app16147174. [Online]. Available: https://www.mdpi.com/2076-3417/16/14/7174.

[22] G. M. Costa, V. T. Santos, P. Alves, A. P. Moreira, and M. R. Petry, "Effects of interface modality on human-robot collaboration: Comparing immersive and non-immersive solutions in disassembly tasks," *IEEE Access*, 2026. DOI: 10.1109/ACCESS.2026.3744466.

[23] M. H. Christoph Petzoldt and M. Freitag, "Review of task allocation for human-robot collaboration in assembly," *International Journal of Computer Integrated Manufacturing*, vol. 36, no. 11, pp. 1675–1715, 2023. DOI: 10.1080/0951192X.2023.2204467.

[24] C. Destouet, H. Tlahig, B. Bettayeb, and B. Mazari, "Flexible job shop scheduling problem under industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement," *Journal of Manufacturing Systems*, vol. 67, pp. 155–173, 2023, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2023.01.004.

[25] E. Lamon, F. Fusaro, E. De Momi, and A. Ajoudani, "A unified architecture for dynamic role allocation and collaborative task planning in mixed human-robot teams," *arXiv preprint arXiv:2301.08038*, 2023. DOI: 10.48550/arXiv.2301.08038.

[26] R. S. Lunding, M. N. Lystbæk, T. Feuchtner, and K. Grønbæk, "Ar-supported human-robot collaboration: Facilitating workspace awareness and parallelized assembly tasks," in *2023 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, IEEE, 2023, pp. 1064–1073. DOI: 10.1109/ISMAR59233.2023.00123.

[27] C. T. Chang and B. Hayes, "A survey of augmented reality for human–robot collaboration," *Machines*, vol. 12, no. 8, p. 540, 2024. DOI: 10.3390/machines12080540.

[28] H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Business & information systems engineering*, vol. 6, no. 4, pp. 239–242, 2014. DOI: 10.1007/s12599-014-0334-4.

[29] A. Rojko, "Industry 4.0 concept: Background and overview.," *International journal of interactive mobile technologies*, vol. 11, no. 5, 2017. DOI: 10.3991/ijim.v11i5.7072.

[30] Á. Segura, H. V. Diez, I. Barandiaran, *et al.*, "Visual computing technologies to support the operator 4.0," *Computers & Industrial Engineering*, vol. 139, p. 105 550, 2020, ISSN: 0360-8352. DOI: 10.1016/j.cie.2018.11.060. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835218306004.

[31] T. van Erp, R. Goncalves, and N. G. M. Rytter, "Design of matrix production systems: A skill-based systems engineering approach," *Procedia CIRP*, vol. 120, pp. 1173–1178, 2023, 56th CIRP International Conference on Manufacturing Systems 2023, ISSN: 2212-8271. DOI: 10.1016/j.procir.2023.09.144.

[32] M. Trierweiler and T. Bauernhansl, "Reconfiguration of production equipment of matrix manufacturing systems," in *Advances in Automotive Production Technology – Theory and Application*, P. Weißgraeber, F. Heieck, and C. Ackermann, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 20–27, ISBN: 978-3-662-62962-8. DOI: 10.1007/978-3-662-62962-8_3.

[33] S. Dechant and H.-C. Möhring, "Development of a capability-based scheduling heuristic for matrix production systems," *Production Engineering*, vol. 20, no. 2, p. 45, 2026. DOI: 10.1007/s11740-026-01424-8.

[34] KUKA, *Matrix production: An example for industrie 4.0*, Web Page, Accessed: 2024-12-20, 2024. [Online]. Available: https://perma.cc/8K4N-WTEM.

[35] Prolynk, *Prolynk - home*, Web Page, Accessed: 2024-12-20, 2024. [Online]. Available: https://www.prolynk.eu/en/acceuil/.

[36] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 5th. Cham: Springer, 2016. DOI: 10.1007/978-3-319-26580-3.

[37] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, "The flexible job shop scheduling problem: A review," *European Journal of Operational Research*, vol. 314, no. 2, pp. 409–432, 2024, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2023.05.017.

[38] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Computers & Operations Research*, vol. 142, p. 105 731, 2022, ISSN: 0305-0548. DOI: 10.1016/j.cor.2022.105731.

[39] L. Bianco, P. Dell'Olmo, and M. Grazia Speranza, "Heuristics for multimode scheduling problems with dedicated resources," *European Journal of Operational Research*, vol. 107, no. 2, pp. 260–271, 1998, ISSN: 0377-2217. DOI: 10.1016/S0377-2217(97)00347-0.

[40] L. Bianco, P. Dell'Olmo, S. Giordani, and M. G. Speranza, "Minimizing makespan in a multimode multiprocessor shop scheduling problem," *Naval Research Logistics (NRL)*, vol. 46, no. 8, pp. 893–911, 1999. DOI: 10.1002/(SICI)1520-6750(199912)46:8<893::AID-NAV2>3.0.CO;2-7.

[41] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and Multidisciplinary Optimization*, vol. 26, no. 6, pp. 369–395, 2004. DOI: 10.1007/s00158-003-0368-6.

[42] L. A. Wolsey, *Integer Programming*, 2nd. Hoboken, NJ: Wiley, 2020. DOI: 10.1002/9781119606475.

[43] M. Padberg and G. Rinaldi, "A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems," *SIAM Review*, vol. 33, no. 1, pp. 60–100, 1991. DOI: 10.1137/1033004.

[44] A. Castro, F. Silva, and V. Santos, "Trends of human-robot collaboration in industry contexts: Handover, learning, and metrics," *Sensors*, vol. 21, no. 12, p. 4113, 2021. DOI: 10.3390/s21124113.

[45] F. Vicentini, "Terminology in safety of collaborative robotics," *Robotics and Computer-Integrated Manufacturing*, vol. 63, p. 101 921, 2020, ISSN: 0736-5845. DOI: 10.1016/j.rcim.2019.101921. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584519300614.

[46] *Robotics – safety requirements – part 1: Industrial robots*, https://www.iso.org/standard/73933.html, Accessed: 2026-03-21, Geneva, Switzerland: International Organization for Standardization, 2025.

[47] J. Schmidtler, V. Knott, C. Hölzel, and K. Bengler, "Human centered assistance applications for the working environment of the future," *Occupational Ergonomics*, vol. 12, no. 3, pp. 83–95, 2015. DOI: 10.3233/OER-150226. eprint: https://journals.sagepub.com/doi/pdf/10.3233/OER-150226. [Online]. Available: https://journals.sagepub.com/doi/abs/10.3233/OER-150226.

[48] S. Thiemermann, *Direkte Mensch-Roboter-Kooperation in der Kleinteilemontage mit einem SCARA-Roboter*. Jost-Jetter Heimsheim, 2005.

[49] D. F. Bortot, "Ergonomic human-robot coexistence in the branch of production," Ph.D. dissertation, Technische Universität München, 2014.

[50] W. Bauer, M. Bender, M. Braun, P. Rally, and O. Scholtz, "Lightweight robots in manual assembly–best to start simply," *Frauenhofer-Institut für Arbeitswirtschaft und Organisation IAO, Stuttgart*, vol. 1, 2016.

[51] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019. DOI: 10.3390/robotics8040100.

[52] O. Danielsson, A. Syberfeldt, R. Brewster, and L. Wang, "Assessing instructions in augmented reality for human-robot collaborative assembly by using demonstrators," *Procedia CIRP*, vol. 63, pp. 89–94, 2017, Manufacturing Systems 4.0 – Proceedings of the 50th CIRP Conference on Manufacturing Systems, ISSN: 2212-8271. DOI: 10.1016/j.procir.2017.02.038. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221282711730135X.

[53] R. Kalpagam Ganesan, Y. K. Rathore, H. M. Ross, and H. Ben Amor, "Better teaming through visual cues: How projecting imagery in a workspace can improve human-robot collaboration," *IEEE Robotics & Automation Magazine*, vol. 25, no. 2, pp. 59–71, 2018. DOI: 10.1109/MRA.2018.2815655.

[54] A. De Franco, E. Lamon, P. Balatti, E. De Momi, and A. Ajoudani, "An intuitive augmented reality interface for task scheduling, monitoring, and work performance improvement in human-robot collaboration," in *2019 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, IEEE, 2019, pp. 75–80. DOI: 10.1109/IWOBI47054.2019.9114472.

[55] A. Hietanen, R. Pieters, M. Lanz, J. Latokartano, and J.-K. Kämäräinen, "Ar-based interaction for human-robot collaborative manufacturing," *Robotics and Computer-Integrated Manufacturing*, vol. 63, p. 101 891, 2020. DOI: 10.1016/j.rcim.2019.101891.

[56] I. Soares, M. Petry, and A. P. Moreira, "Programming robots by demonstration using augmented reality," *Sensors*, vol. 21, no. 17, p. 5976, 2021. DOI: 10.3390/s21175976.

[57] D. Bambuŝek, Z. Materna, M. Kapinus, V. Beran, and P. Smrž, "Combining interactive spatial augmented reality with head-mounted display for end-user collaborative robot programming," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, 2019, pp. 1–8. DOI: 10.1109/RO-MAN46459.2019.8956315.

[58] A. Syberfeldt, O. Danielsson, M. Holm, and L. Wang, "Visual assembling guidance using augmented reality," *Procedia Manufacturing*, vol. 1, pp. 98–109, 2015, 43rd North American Manufacturing Research Conference, NAMRC 43, 8-12 June 2015, UNC Charlotte, North Carolina, United States, ISSN: 2351-9789. DOI: 10.1016/j.promfg.2015.

09.068. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978915010689.

[59] P. Milgram and F. Kishino, "A taxonomy of mixed reality visual displays," *IEICE TRANSACTIONS on Information and Systems*, vol. 77, no. 12, pp. 1321–1329, 1994. DOI: 10.1587/e77-d_12_1321.

[60] *Mixed and augmented reality (mar) reference model*, https://www.iso.org/standard/30824.html, Accessed: 2026-03-21, Geneva, Switzerland: International Organization for Standardization, 2019.

[61] P. A. Rauschnabel, R. Felix, C. Hinsch, H. Shahab, and F. Alt, "What is xr? towards a framework for augmented and virtual reality," *Computers in Human Behavior*, vol. 133, p. 107 289, 2022, ISSN: 0747-5632. DOI: 10.1016/j.chb.2022.107289. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074756322200111X.

[62] R. T. Azuma, "A survey of augmented reality," *Presence: teleoperators & virtual environments*, vol. 6, no. 4, pp. 355–385, 1997. DOI: 10.1162/pres.1997.6.4.355.

[63] M. Speicher, B. D. Hall, and M. Nebeling, "What is mixed reality?" In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, ser. CHI '19, Glasgow, Scotland Uk: Association for Computing Machinery, 2019, pp. 1–15, ISBN: 9781450359702. DOI: 10.1145/3290605.3300767. [Online]. Available: https://doi.org/10.1145/3290605.3300767.

[64] M. J. Maas and J. M. Hughes, "Virtual, augmented and mixed reality in k–12 education: A review of the literature," *Technology, Pedagogy and Education*, vol. 29, no. 2, pp. 231–249, 2020. DOI: 10.1080/1475939X.2020.1737210. eprint: https://doi.org/10.1080/1475939X.2020.1737210. [Online]. Available: https://doi.org/10.1080/1475939X.2020.1737210.

[65] D. Drascic and P. Milgram, "Perceptual issues in augmented reality," in *Stereoscopic displays and virtual reality systems III*, Spie, vol. 2653, 1996, pp. 123–134. DOI: 10.1117/12.237425.

[66] C.-Y. Chang, C.-L. D. Chena, and W.-K. Chang, "Research on immersion for learning using virtual reality, augmented reality and mixed reality," *Enfance*, vol. 3, no. 3, pp. 413–426, 2019.

[67] Å. Fast-Berglund, L. Gong, and D. Li, "Testing and validating extended reality (xr) technologies in manufacturing," *Procedia Manufacturing*, vol. 25, pp. 31–38, 2018, Proceedings of the 8th Swedish Production Symposium (SPS 2018), ISSN: 2351-9789. DOI: 10.1016/j.promfg.2018.06.054. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978918305730.

[68] S. Alizadehsalehi, A. Hadavi, and J. C. Huang, "From bim to extended reality in aec industry," *Automation in Construction*, vol. 116, p. 103 254, 2020, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2020.103254. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580519315146.

[69] T. Masood and J. Egger, "Adopting augmented reality in the age of industrial digitalisation," *Computers in Industry*, vol. 115, p. 103 112, 2020, ISSN: 0166-3615. DOI: 10.1016/j.compind.2019.07.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361519301186.

[70] T. Masood and J. Egger, "Augmented reality in support of industry 4.0—implementation challenges and success factors," *Robotics and Computer-Integrated Manufacturing*, vol. 58, pp. 181–195, 2019, ISSN: 0736-5845. DOI: 10.1016/j.rcim.2019.02.003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0736584518304101.

[71] T. Bakale and R. Picek, "Application of augmented reality in creating added value for an erp system," in *2023 4th International Conference on Communications, Information, Electronic and Energy Systems (CIEES)*, 2023, pp. 1–5. DOI: 10.1109/CIEES58940.2023.10378765.

[72] Z.-H. Lai, W. Tao, M. C. Leu, and Z. Yin, "Smart augmented reality instructional system for mechanical assembly towards worker-centered intelligent manufacturing," *Journal of Manufacturing Systems*, vol. 55, pp. 69–81, 2020, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2020.02.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612520300303.

[73] D. Aschenbrenner, F. Leutert, A. Çençen, *et al.*, "Comparing human factors for augmented reality supported single-user and collaborative repair operations of industrial robots," *Frontiers in Robotics and AI*, vol. 6, p. 37, 2019. DOI: 10.3389/frobt.2019.00037.

[74] D. Fang, H. Xu, X. Yang, and M. Bian, "An augmented reality-based method for remote collaborative real-time assistance: From a system perspective," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 412–425, 2020. DOI: 10.1007/s11036-019-01244-4.

[75] W. Wang, F. Wang, W. Song, and S. Su, "Application of augmented reality (ar) technologies in inhouse logistics," in *E3S Web of Conferences*, EDP Sciences, vol. 145, 2020, p. 02 018. DOI: 10.1051/e3sconf/202014502018.

[76] N. Gavish, T. Gutiérrez, S. Webel, *et al.*, "Evaluating virtual reality and augmented reality training for industrial maintenance and assembly tasks," *Interactive Learning Environments*, vol. 23, no. 6, pp. 778–798, 2015. DOI: 10.1080/10494820.2013.815221. eprint: https://doi.org/10.1080/10494820.2013.815221. [Online]. Available: https://doi.org/10.1080/10494820.2013.815221.

[77] D. Segovia, M. Mendoza, E. Mendoza, and E. González, "Augmented reality as a tool for production and quality monitoring," *Procedia Computer Science*, vol. 75, pp. 291–300, 2015, 2015 International Conference Virtual and Augmented Reality in Education, ISSN: 1877-0509. DOI: 10.1016/j.procs.2015.12.250. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915037114.

[78] R. C. Nascimento, J. G. Martins, D. G. Gonzalez, *et al.*, "Augmented reality and deep learning-based framework for defect detection in reflective parts," in *2026 12th International Conference on Automation, Robotics and Applications (ICARA)*, 2026, pp. 368–374. DOI: 10.1109/ICARA69401.2026.11480351.

[79] C. Kollatsch, M. Schumann, P. Klimant, V. Wittstock, and M. Putz, "Mobile augmented reality based monitoring of assembly lines," *Procedia CIRP*, vol. 23, pp. 246–251, 2014, 5th CATS 2014 - CIRP Conference on Assembly Technologies and Systems, ISSN: 2212-8271. DOI: 10.1016/j.procir.2014.10.100. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827114011573.

[80] A. Doshi, R. T. Smith, B. H. Thomas, and C. Bouras, "Use of projector based augmented reality to improve manual spot-welding precision and accuracy for automotive manufacturing," *The International Journal of Advanced Manufacturing Technology*, vol. 89, no. 5, pp. 1279–1293, 2017. DOI: 10.1007/s00170-016-9164-5.

[81] P. Reipschläger and R. Dachselt, "Designar: Immersive 3d-modeling combining augmented reality with interactive displays," in *Proceedings of the 2019 ACM International Conference on Interactive Surfaces and Spaces*, ser. ISS '19, Daejeon, Republic of Korea: Association for Computing Machinery, 2019, pp. 29–41, ISBN: 9781450368919. DOI: 10.1145/3343055.3359718. [Online]. Available: https://doi.org/10.1145/3343055.3359718.

[82] D. Tatić and B. Tešić, "The application of augmented reality technologies for the improvement of occupational safety in an industrial environment," *Computers in Industry*, vol. 85, pp. 1–10, 2017, ISSN: 0166-3615. DOI: 10.1016/j.compind.2016.11.004. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361516302718.

[83] J. Egger and T. Masood, "Augmented reality in support of intelligent manufacturing – a systematic literature review," *Computers & Industrial Engineering*, vol. 140, p. 106 195, 2020, ISSN: 0360-8352. DOI: 10.1016/j.cie.2019.106195. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360835219306643.

[84] P. Fraga-Lamas, T. M. FernáNdez-CaraméS, Ó. Blanco-Novoa, and M. A. Vilar-Montesinos, "A review on industrial augmented reality systems for the industry 4.0 shipyard," *IEEE Access*, vol. 6, pp. 13 358–13 375, 2018. DOI: 10.1109/ACCESS.2018.2808326.

[85] J. Miller, "Overcoming the limitations of commodity augmented reality head mounted displays for use in product assembly," M.S. thesis, Iowa State University, 2019.

[86] B. M. Khuong, K. Kiyokawa, A. Miller, J. J. La Viola, T. Mashita, and H. Takemura, "The effectiveness of an ar-based context-aware assembly support system in object assembly," in *2014 IEEE virtual reality (VR)*, IEEE, 2014, pp. 57–62. DOI: 10.1109/VR.2014.6802051.

[87] B. J. Park, S. J. Hunt, G. J. Nadolski, and T. P. Gade, "Augmented reality improves procedural efficiency and reduces radiation dose for ct-guided lesion targeting: A phantom study using hololens 2," *Scientific reports*, vol. 10, no. 1, p. 18 620, 2020. DOI: 10.1038/s41598-020-75676-4.

[88] C. Gsaxner, J. Li, A. Pepe, D. Schmalstieg, and J. Egger, "Inside-out instrument tracking for surgical navigation in augmented reality," in *Proceedings of the 27th ACM Symposium on Virtual Reality Software and Technology*, ser. VRST '21, Osaka, Japan: Association for Computing Machinery, 2021, ISBN: 9781450390927. DOI: 10.1145/3489849.3489863. [Online]. Available: https://doi.org/10.1145/3489849.3489863.

[89] M. Doughty and N. R. Ghugre, "Head-mounted display-based augmented reality for image-guided media delivery to the heart: A preliminary investigation of perceptual accuracy," *Journal of imaging*, vol. 8, no. 2, p. 33, 2022. DOI: 10.3390/jimaging8020033.

[90] X. Hu, F. R. Y. Baena, and F. Cutolo, "Alignment-free offline calibration of commercial optical see-through head-mounted displays with simplified procedures," *IEEE Access*, vol. 8, pp. 223 661–223 674, 2020. DOI: 10.1109/ACCESS.2020.3044184.

[91] N. Guo, T. Wang, B. Yang, L. Hu, H. Liu, and Y. Wang, "An online calibration method for microsoft hololens," *IEEE Access*, vol. 7, pp. 101 795–101 803, 2019. DOI: 10.1109/ACCESS.2019.2930701.

[92] D. Khan, S. Ullah, D.-M. Yan, *et al.*, "Robust tracking through the design of high quality fiducial markers: An optimization tool for artoolkit," *IEEE Access*, vol. 6, pp. 22 421–22 433, 2018. DOI: 10.1109/ACCESS.2018.2801028.

[93] S. Garrido-Jurado, J. Garrido, D. Jurado-Rodríguez, F. Vázquez, and R. Muñoz-Salinas, "Reflection-aware generation and identification of square marker dictionaries," *Sensors*, vol. 22, no. 21, p. 8548, 2022. DOI: 10.3390/s22218548.

[94] Ó. Blanco-Novoa, T. M. FernáNdez-CaraméS, P. Fraga-Lamas, and M. A. Vilar-Montesinos, "A practical evaluation of commercial industrial augmented reality systems in an industry 4.0 shipyard," *IEEE Access*, vol. 6, pp. 8201–8218, 2018. DOI: 10.1109/ACCESS.2018.2802699.

[95] P. Q. Brito and J. Stoyanova, "Marker versus markerless augmented reality. which has more impact on users?" *International Journal of Human–Computer Interaction*, vol. 34, no. 9, pp. 819–833, 2018. DOI: 10.1080/10447318.2017.1393974. eprint: https://doi.org/10.1080/10447318.2017.1393974. [Online]. Available: https://doi.org/10.1080/10447318.2017.1393974.

[96] J. Vanherck, B. Zoomers, L. Jorissen, I. Vandebroeck, E. Joris, and N. Michiels, "Large-area spatially aligned anchors," in *Extended Reality*, L. T. De Paolis, P. Arpaia, and M. Sacco, Eds., Cham: Springer Nature Switzerland, 2024, pp. 42–60, ISBN: 978-3-031-71707-9. DOI: 10.1007/978-3-031-71707-9_3.

[97] A. Blaga and L. Tamas, "Augmented reality for digital manufacturing," in *2018 26th Mediterranean Conference on Control and Automation (MED)*, 2018, pp. 173–178. DOI: 10.1109/MED.2018.8443028.

[98] S. Park, S. Bokijonov, and Y. Choi, "Review of microsoft hololens applications over the past five years," *Applied sciences*, vol. 11, no. 16, p. 7259, 2021. DOI: 10.3390/app11167259.

[99] A. Thabit, W. J. Niessen, E. B. Wolvius, and T. Van Walsum, "Evaluation of marker tracking using mono and stereo vision in microsoft hololens for surgical navigation," in *Medical Imaging 2022: Image-Guided Procedures, Robotic Interventions, and Modeling*, SPIE, vol. 12034, 2022, pp. 253–262. DOI: 10.1117/12.2607262.

[100] A. Sadhu, J. E. Peplinski, A. Mohammadkhorasani, and F. Moreu, "A review of data management and visualization techniques for structural health monitoring using bim and virtual or augmented reality," *Journal of Structural Engineering*, vol. 149, no. 1, p. 03 122 006, 2023. DOI: 10.1061/(ASCE)ST.1943-541X.0003498. eprint: https://ascelibrary.org/doi/pdf/10.1061/%28ASCE%29ST.1943-541X.0003498. [Online]. Available: https://ascelibrary.org/doi/abs/10.1061/%28ASCE%29ST.1943-541X.0003498.

[101] A. Mohammadkhorasani, K. Malek, R. Mojidra, *et al.*, "Augmented reality-computer vision combination for automatic fatigue crack detection and localization," *Computers in Industry*, vol. 149, p. 103 936, 2023, ISSN: 0166-3615. DOI: 10.1016/j.compind.2023.103936. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0166361523000866.

[102] P. Hübner, K. Clintworth, Q. Liu, M. Weinmann, and S. Wursthorn, "Evaluation of hololens tracking and depth sensing for indoor mapping applications," *Sensors*, vol. 20, no. 4, p. 1021, 2020. DOI: 10.3390/s20041021.

[103] D. Puljiz, F. Krebs, F. Bosing, and B. Hein, "What the hololens maps is your workspace: Fast mapping and set-up of robot cells via head mounted displays and augmented reality," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 11 445–11 451. DOI: 10.1109/IROS45743.2020.9340879.

[104] M. Quandt, B. Knoke, C. Gorldt, M. Freitag, and K.-D. Thoben, "General requirements for industrial augmented reality applications," *Procedia CIRP*, vol. 72, pp. 1130–1135, 2018, 51st CIRP Conference on Manufacturing Systems, ISSN: 2212-8271. DOI: 10.1016/j.procir.2018.03.061. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221282711830163X.

[105] J. Jetter, J. Eimecke, and A. Rese, "Augmented reality tools for industrial applications: What are potential key performance indicators and who benefits?" *Computers in Human Behavior*, vol. 87, pp. 18–33, 2018, ISSN: 0747-5632. DOI: 10.1016/j.chb.2018.04.054. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S074756321830222X.

[106] E. Bottani and G. Vignali, "Augmented reality technology in the manufacturing industry: A review of the last decade," *IISE Transactions*, vol. 51, no. 3, pp. 284–310, 2019. DOI: 10.1080/24725854.2018.1493244. eprint: https://doi.org/10.1080/24725854.2018.1493244. [Online]. Available: https://doi.org/10.1080/24725854.2018.1493244.

[107] A. Türkyılmaz, Ö. Şenvar, I. Ünal, and S. Bulkan, "A research survey: Heuristic approaches for solving multi objective flexible job shop problems," *Journal of Intelligent Manufacturing*, vol. 31, no. 8, pp. 1949–1983, Feb. 2020, ISSN: 1572-8145. DOI: 10.1007/s10845-020-01547-4.

[108] A. Azzouz, M. Ennigrou, and L. B. Said, "A self-adaptive hybrid algorithm for solving flexible job-shop problem with sequence dependent setup time," *Procedia Computer Science*, vol. 112, pp. 457–466, 2017, Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES-20176-8 September 2017, Marseille, France, ISSN: 1877-0509. DOI: 10.1016/j.procs.2017.08.023.

[109] D. B. Fontes, S. M. Homayouni, and J. F. Gonçalves, "A hybrid particle swarm optimization and simulated annealing algorithm for the job shop scheduling problem with transport resources," *European Journal of Operational Research*, vol. 306, no. 3, pp. 1140–1157, 2023, ISSN: 0377-2217. DOI: 10.1016/j.ejor.2022.09.006.

[110] H. Tang, B. Fang, R. Liu, Y. Li, and S. Guo, "A hybrid teaching and learning-based optimization algorithm for distributed sand casting job-shop scheduling problem," *Applied Soft Computing*, vol. 120, p. 108 694, 2022, ISSN: 1568-4946. DOI: 10.1016/j.asoc.2022.108694.

[111] K. Z. Gao, P. N. Suganthan, Q. K. Pan, M. F. Tasgetiren, and A. Sadollah, "Artificial bee colony algorithm for scheduling and rescheduling fuzzy flexible job shop problem with new job insertion," *Knowledge-Based Systems*, vol. 109, pp. 1–16, 2016, ISSN: 0950-7051. DOI: 10.1016/j.knosys.2016.06.014.

[112] Z. Zhu and X. Zhou, "Flexible job-shop scheduling problem with job precedence constraints and interval grey processing time," *Computers & Industrial Engineering*, vol. 149, p. 106 781, 2020, ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106781.

[113] V. Boyer, J. Vallikavungal, X. Cantú Rodríguez, and M. A. Salazar-Aguilar, "The generalized flexible job shop scheduling problem," *Computers & Industrial Engineering*, vol. 160, p. 107 542, 2021, ISSN: 0360-8352. DOI: 10.1016/j.cie.2021.107542.

[114] Z. Zhao, S. Liu, M. Zhou, D. You, and X. Guo, "Heuristic scheduling of batch production processes based on petri nets and iterated greedy algorithms," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 1, pp. 251–261, 2022. DOI: 10.1109/TASE.2020.3027532.

[115] W. Shao, Z. Shao, and D. Pi, "Lot sizing and scheduling problem in distributed heterogeneous hybrid flow shop and learning-driven iterated local search algorithm," *IEEE Transactions on Automation Science and Engineering*, vol. 21, no. 4, pp. 6483–6497, 2024. DOI: 10.1109/TASE.2023.3326301.

[116] M. A. Salido, J. Escamilla, A. Giret, and F. Barber, "A genetic algorithm for energy-efficiency in job-shop scheduling," *The International Journal of Advanced Manufacturing Technology*, vol. 85, no. 5–8, pp. 1303–1314, Nov. 2015, ISSN: 1433-3015. DOI: 10.1007/s00170-015-7987-0.

[117] M. A. Salido, J. Escamilla, F. Barber, and A. Giret, "Rescheduling in job-shop problems for sustainable manufacturing systems," *Journal of Cleaner Production*, vol. 162, S121–S132, 2017, Innovative Products and Services for Sustainable Societal Development, ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2016.11.002.

[118] Y. Li, Y. He, Y. Wang, F. Tao, and J. W. Sutherland, "An optimization method for energy-conscious production in flexible machining job shops with dynamic job arrivals and machine breakdowns," *Journal of Cleaner Production*, vol. 254, p. 120 009, 2020, ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2020.120009.

[119] M. Dai, D. Tang, A. Giret, and M. A. Salido, "Multi-objective optimization for energy-efficient flexible job shop scheduling problem with transportation constraints," *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 143–157, 2019, ISSN: 0736-5845. DOI: 10.1016/j.rcim.2019.04.006.

[120] Q. Liu, M. Zhan, F. O. Chekem, X. Shao, B. Ying, and J. W. Sutherland, "A hybrid fruit fly algorithm for solving flexible job-shop scheduling to reduce manufacturing carbon footprint," *Journal of Cleaner Production*, vol. 168, pp. 668–678, 2017, ISSN: 0959-6526. DOI: 10.1016/j.jclepro.2017.09.037.

[121] W. Shao, D. Pi, and Z. Shao, "A pareto-based estimation of distribution algorithm for solving multiobjective distributed no-wait flow-shop scheduling problem with sequence-dependent setup time," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 3, pp. 1344–1360, 2019. DOI: 10.1109/TASE.2018.2886303.

[122] W. Cheng, C. Zhang, L. Meng, Y. Ren, and B. Zhang, "Multi-threaded constraint programming-assisted space perception evolutionary algorithm for energy-efficient flexible job shop scheduling problem with sequence-dependent setup times," *Computers & Operations Research*, vol. 192, p. 107 494, 2026, ISSN: 0305-0548. DOI: 10.1016/j.cor.2026.107494. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0305054826001127.

[123] M. Chang, L. Tang, and S. Zhao, "A reinforcement learning-based lagrangian decomposition approach for energy-oriented scheduling optimization in steelmaking process," *IEEE Transactions on Automation Science and Engineering*, vol. 23, pp. 3843–3862, 2026. DOI: 10.1109/TASE.2025.3623447.

[124] M. Ben Hamida, A. Azzouz, and L. Ben Said, "A self-learning variable neighborhood search algorithm based on reinforcement learning for green flexible job-shop scheduling problem: M. ben hamida et al.," *Journal of Heuristics*, vol. 32, no. 2, p. 17, 2026. DOI: 10.1007/s10732-026-09590-9.

[125] E. Merlo, E. Lamon, F. Fusaro, *et al.*, "An ergonomic role allocation framework for dynamic human–robot collaborative tasks," *Journal of Manufacturing Systems*, vol. 67, pp. 111–121, 2023, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2022.12.011.

[126] A. Raatz, S. Blankemeyer, T. Recker, D. Pischke, and P. Nyhuis, "Task scheduling method for hrc workplaces based on capabilities and execution time assumptions for robots," *CIRP Annals*, vol. 69, no. 1, pp. 13–16, 2020, ISSN: 0007-8506. DOI: 10.1016/j.cirp.2020.04.030.

[127] M. C. Z. N. Gjeldum A. Aljinovic and M. Mladineo, "Collaborative robot task allocation on an assembly line using the decision support system," *International Journal of Computer Integrated Manufacturing*, vol. 35, no. 4-5, pp. 510–526, 2022. DOI: 10.1080/0951192X.2021.1946856.

[128] H.-y. Liao, Y. Chen, B. Hu, and S. Behdad, "Optimization-based disassembly sequence planning under uncertainty for human–robot collaboration," *Journal of Mechanical Design*, vol. 145, no. 2, p. 022 001, Nov. 2022, ISSN: 1050-0472. DOI: 10.1115/1.4055901. eprint: https://asmedigitalcollection.asme.org/mechanicaldesign/article-pdf/145/2/022001/6934385/md_145_2_022001.pdf. [Online]. Available: https://doi.org/10.1115/1.4055901.

[129] "Human-robot collaboration for e-waste disassembly: Task modeling, allocation, and ergonomic assessment," *Ergonomics*, vol. 0, no. 0, pp. 1–17, 2026, PMID: 41902609. DOI: 10.1080/00140139.2026.2650455. eprint: https://doi.org/10.1080/00140139.2026.2650455. [Online]. Available: https://doi.org/10.1080/00140139.2026.2650455.

[130] E. Lamon, A. De Franco, L. Peternel, and A. Ajoudani, "A capability-aware role allocation approach to industrial assembly tasks," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3378–3385, 2019. DOI: 10.1109/LRA.2019.2926963.

[131] A. Casalino, A. M. Zanchettin, L. Piroddi, and P. Rocco, "Optimal scheduling of human–robot collaborative assembly operations with time petri nets," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 1, pp. 70–84, 2021. DOI: 10.1109/TASE.2019.2932150.

[132] F. Yu, C. Lu, L. Yin, and B. Zhang, "A self-learning memetic algorithm for human-robot collaboration scheduling in energy-efficient distributed mixed fuzzy welding shop," *IEEE Transactions on Automation Science and Engineering*, vol. 22, pp. 6595–6607, 2025. DOI: 10.1109/TASE.2024.3448435.

[133] J. G. Martins, G. M. Costa, M. R. Petry, P. Costa, and A. P. Moreira, "A multi-user multi-robot collaboration through augmented reality," in *2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2024, pp. 144–150. DOI: 10.1109/ICARSC61747.2024.10535948.

[134] S. Blankemeyer, R. Wiemann, L. Posniak, C. Pregizer, and A. Raatz, "Intuitive robot programming using augmented reality," *Procedia CIRP*, vol. 76, pp. 155–160, 2018, 7th CIRP Conference on Assembly Technologies and Systems (CATS 2018), ISSN: 2212-8271. DOI: 10.1016/j.procir.2018.02.028. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2212827118300933.

[135] V. H. T. dos Santos, "Human-robot collaborative workspaces supported by ar head-mounted displays," 2025.

[136] A. Ajikumar, S. Marsella, and M. Moghaddam, "Do you understand me, dum-e? enabling seamless human-robot communication through augmented reality," in *2024 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, IEEE, 2024, pp. 529–532. DOI: 10.1109/ISMAR-Adjunct64951.2024.00153.

[137] A. Madani, P. P. Niaz, B. Guler, Y. Aydin, and C. Basdogan, "Robot-assisted drilling on curved surfaces with haptic guidance under adaptive admittance control," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2022, pp. 3723–3730. DOI: 10.1109/IROS47612.2022.9982000.

[138] S. K. Tadeja, T. Zhou, M. Capponi, K. Walas, T. Bohné, and F. Forni, "Using augmented reality in human-robot assembly: A comparative study of eye-gaze and hand-ray pointing methods," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, pp. 8786–8793. DOI: 10.1109/IROS58592.2024.10802282.

[139] A. Sanna, F. Manuri, J. Fiorenza, and F. De Pace, "Bari: An affordable brain-augmented reality interface to support human–robot collaboration in assembly tasks," *Information*, vol. 13, no. 10, p. 460, 2022. DOI: 10.3390/info13100460.

[140] N. Dimitropoulos, T. Togias, N. Zacharaki, G. Michalos, and S. Makris, "Seamless human–robot collaborative assembly using artificial intelligence and wearable devices," *Applied Sciences*, vol. 11, no. 12, p. 5699, 2021. DOI: 10.3390/app11125699.

[141] Y. Tong, Q. Zhang, F. M. Galeano, and Z. Ji, "Ar and hrc integration for enhanced pragmatic quality," in *2024 IEEE International Conference on Industrial Technology (ICIT)*, IEEE, 2024, pp. 1–6. DOI: 10.1109/ICIT58233.2024.10540971.

[142] G. Apostolopoulos, D. Andronas, N. Fourtakas, and S. Makris, "Operator training framework for hybrid environments: An augmented reality module using machine learning object recognition," *Procedia CIRP*, vol. 106, pp. 102–107, 2022. DOI: 10.1016/j.procir.2022.02.162.

[143] D. Andronas, G. Apostolopoulos, N. Fourtakas, and S. Makris, "Multi-modal interfaces for natural human-robot interaction," *Procedia Manufacturing*, vol. 54, pp. 197–202, 2021. DOI: 10.1016/j.promfg.2021.07.030.

[144] M. Dalle Mura and G. Dini, "A proposal of an assembly workstation for car panel fitting aided by an augmented reality device," *Procedia CIRP*, vol. 103, pp. 225–230, 2021. DOI: 10.1016/j.procir.2021.10.036.

[145] A. A. Tadesse, K.-J. Wang, C. Lin, and Y.-L. Hsu, "A human-robot collaboration model empowered by object detection-driven ar assistance and operator behavior prediction," *International Journal of Computer Integrated Manufacturing*, vol. 0, no. 0, pp. 1–21, 2026. DOI: 10.1080/0951192X.2026.2649614. eprint: https://doi.org/10.1080/0951192X.2026.2649614. [Online]. Available: https://doi.org/10.1080/0951192X.2026.2649614.

[146] H. Xu, L. Zhou, X. Zhou, and P. Zheng, "An augmented reality-enhanced large language model-based interactive system for human-robot collaboration," *Available at SSRN 6511489*, DOI: 10.2139/ssrn.6511489.

[147] C. Liu, D. Tang, and L. Zhang, "Probing an augmented reality-assisted human-robot collaborative assembly approach based on deep reinforcement learning," in *2024 International Conference on Automation in Manufacturing, Transportation and Logistics (ICaMaL)*, IEEE, 2024, pp. 1–5. DOI: 10.1109/ICaMaL62577.2024.10919818.

[148] C. Liu, D. Tang, H. Zhu, Z. Zhang, L. Wang, and Q. Nie, "Ar-assisted human-robot collaborative assembly system: Integrating visual language model and deep reinforcement learning for task planning and seamless interactive guidance," *Journal of Manufacturing Systems*, vol. 84, pp. 40–67, 2026, ISSN: 0278-6125. DOI: 10.1016/j.jmsy.2025.11.019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0278612525002791.

[149] J. Wang, Y. Yan, Y. Hu, X. Yang, and L. Zhang, "A transfer reinforcement learning and digital-twin based task allocation method for human-robot collaboration assembly," *Engineering Applications of Artificial Intelligence*, vol. 144, p. 110 064, 2025. DOI: 10.1016/j.engappai.2025.110064.

[150] N. Rogeau and A. Rezaei Rad, "Collaborative timber joint assembly: Augmented reality for multi-level human-robot interaction," *International Journal of Architectural Computing*, vol. 22, no. 4, pp. 642–660, 2024. DOI: 10.1177/14780771241286605.

[151] D. Ungureanu, F. Bogo, S. Galliani, *et al.*, "Hololens 2 research mode as a tool for computer vision research," *arXiv preprint arXiv:2008.11239*, 2020. DOI: 10.48550/arXiv.2008.11239.

[152] R. Radkowski and S. Kanunganti, "Augmented reality system calibration for assembly support with the microsoft hololens," ser. International Manufacturing Science and Engineering Conference, vol. Volume 3: Manufacturing Equipment and Systems, Jun. 2018, V003T02A021. DOI: 10.1115/MSEC2018-6660. eprint: https://asmedigitalcollection.asme.org/MSEC/proceedings-pdf/MSEC2018/51371/V003T02A021/2520178/v003t02a021-msec2018-6660.pdf. [Online]. Available: https://doi.org/10.1115/MSEC2018-6660.

[153] R. Vassallo, A. Rankin, E. C. Chen, and T. M. Peters, "Hologram stability evaluation for microsoft hololens," in *Medical Imaging 2017: Image Perception, Observer Performance, and Technology Assessment*, Spie, vol. 10136, 2017, pp. 295–300. DOI: 10.1117/12.2255831.

[154] Microsoft, *Microsoft hololens | mixed reality technology for busines*, Web Page, Accessed: 2026-02-18, 2022. [Online]. Available: https://learn.microsoft.com/en-us/hololens/.

[155] Microsoft, *Qr code tracking overview*, Web Page, Accessed: 2026-02-19, 2024. [Online]. Available: https://learn.microsoft.com/en-us/windows/mixed-reality/develop/advanced-concepts/qr-code-tracking-overview.

[156] L. Qian, A. Deguet, and P. Kazanzides, "Arssist: Augmented reality on a head-mounted display for the first assistant in robotic surgery," *Healthcare Technology Letters*, vol. 5, no. 5, pp. 194–200, 2018. DOI: 10.1049/htl.2018.5065. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/htl.2018.5065. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/htl.2018.5065.

[157] M. Brand, L. A. Wulff, Y. Hamdani, and T. Schüppstuhl, "Accuracy of marker tracking on an optical see-through head mounted display," in *Annals of Scientific Society for Assembly, Handling and Industrial Robotics*, T. Schüppstuhl, K. Tracht, and D. Henrich, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 21–31, ISBN: 978-3-662-61755-7. DOI: 10.1007/978-3-662-61755-7_3.

[158] S. Eom, S. Kim, S. Rahimpour, and M. Gorlatova, "Ar-assisted surgical guidance system for ventriculostomy," in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, 2022, pp. 402–405. DOI: 10.1109/VRW55335.2022.00087.

[159] T. Frantz, B. Jansen, J. Duerinck, and J. Vandemeulebroucke, "Augmenting microsoft's hololens with vuforia tracking for neuronavigation," *Healthcare Technology Letters*, vol. 5, no. 5, pp. 221–225, 2018. DOI: 10.1049/htl.2018.5079. eprint: https://ietresearch.onlinelibrary.wiley.com/doi/pdf/10.1049/htl.2018.5079. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/htl.2018.5079.

[160] A. H. Kyaw, A. H. Xu, G. Jahn, N. van den Berg, C. Newnham, and S. Zivkovic, "Augmented reality for high precision fabrication of glued laminated timber beams," *Automation in Construction*, vol. 152, p. 104 912, 2023, ISSN: 0926-5805. DOI: 10.1016/j.autcon.2023.104912. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0926580523001723.

[161] R. Arrais, G. Veiga, T. T. Ribeiro, *et al.*, "Application of the open scalable production system to machine tending of additive manufacturing operations by a mobile manipulator," in *Progress in Artificial Intelligence*, P. Moura Oliveira, P. Novais, and L. P. Reis, Eds., Cham: Springer International Publishing, 2019, pp. 345–356, ISBN: 978-3-030-30244-3. DOI: 10.1007/978-3-030-30244-3_29.

[162] S. Hopko, J. Wang, and R. Mehta, "Human factors considerations and metrics in shared space human-robot collaboration: A systematic review," *Frontiers in Robotics and AI*, vol. 9, p. 799 522, 2022. DOI: 10.3389/frobt.2022.799522.

[163] Y. Liu, G. Caldwell, M. Rittenbruch, M. Belek Fialho Teixeira, A. Burden, and M. Guertler, "What affects human decision making in human–robot collaboration?: A scoping review," *Robotics*, vol. 13, no. 2, p. 30, 2024. DOI: 10.3390/robotics13020030.

[164] H. Arnarson and M. Sæterbø, "Reconfigurable manufacturing systems: A visual simulation approach in matrix manufacturing," in *Advanced Manufacturing and Automation XIII*, Y. Wang, T. Yu, and K. Wang, Eds., Springer, Singapore: Springer Nature Singapore, 2024, pp. 144–150, ISBN: 978-981-97-0665-5. DOI: 10.1007/978-981-97-0665-5_19.

[165] P. K. R. Maddikunta, Q.-V. Pham, N. Deepa, *et al.*, "Industry 5.0: A survey on enabling technologies and potential applications," *Journal of industrial information integration*, vol. 26, p. 100 257, 2022. DOI: 10.1016/j.jii.2021.100257.

[166] M. R. Endsley, "Toward a theory of situation awareness in dynamic systems," *Human Factors*, vol. 37, no. 1, pp. 32–64, 1995. DOI: 10.1518/001872095779049543.

[167] A. D. Dragan, K. C. T. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Proc. 8th ACM/IEEE Int. Conf. Human-Robot Interaction (HRI)*, 2013, pp. 301–308. DOI: 10.1109/HRI.2013.6483603.

[168] E. Rosen, D. Whitney, E. Phillips, *et al.*, "Communicating and controlling robot arm motion intent through mixed-reality head-mounted displays," *The International Journal of Robotics Research*, vol. 38, no. 12–13, pp. 1513–1526, 2019. DOI: 10.1177/0278364919842925.

[169] M. Walker, H. Hedayati, J. Lee, and D. Szafir, "Communicating robot motion intent with augmented reality," in *Proc. ACM/IEEE Int. Conf. Human-Robot Interaction (HRI)*, 2018, pp. 316–324. DOI: 10.1145/3171221.3171253.

[170] R. Suzuki, A. Karim, T. Xia, H. Hedayati, and N. Marquardt, "Augmented reality and robotics: A survey and taxonomy for AR-enhanced human-robot interaction and robotic interfaces," in *Proc. CHI Conf. Human Factors in Computing Systems*, 2022. DOI: 10.1145/3491102.3517719.

[171] M. Pascher, U. Gruenefeld, S. Schneegass, and J. Gerken, "How to communicate robot motion intent: A scoping review," in *Proc. CHI Conf. Human Factors in Computing Systems (CHI '23)*, ACM, 2023, 409:1–409:17. DOI: 10.1145/3544548.3580857.

[172] G. Hoffman, "Evaluating fluency in human–robot collaboration," *IEEE Transactions on Human-Machine Systems*, vol. 49, no. 3, pp. 209–218, 2019. DOI: 10.1109/THMS.2019.2904558.

[173] P. Chandler and J. Sweller, "Cognitive load theory and the format of instruction," *Cognition and Instruction*, vol. 8, no. 4, pp. 293–332, 1991. DOI: 10.1207/s1532690xci0804_2.

[174] C. D. Wickens and C. M. Carswell, "The proximity compatibility principle: Its psychological foundation and relevance to display design," *Human Factors*, vol. 37, no. 3, pp. 473–494, 1995. DOI: 10.1518/001872095779049408.

[175] D. M. Hoffman, A. R. Girshick, K. Akeley, and M. S. Banks, "Vergence–accommodation conflicts hinder visual performance and cause visual fatigue," *Journal of Vision*, vol. 8, no. 3, p. 33, 2008. DOI: 10.1167/8.3.33.

[176] T. Wang, P. Zheng, S. Li, and L. Wang, "Multimodal human–robot interaction for human-centric smart manufacturing: A survey," *Advanced Intelligent Systems*, vol. 6, no. 3, p. 2 300 359, 2024. DOI: 10.1002/aisy.202300359.

[177] M. Dhanda, B. A. Rogers, S. Hall, E. Dekoninck, and V. Dhokia, "Reviewing human-robot collaboration in manufacturing: Opportunities and challenges in the context of industry 5.0," *Robotics and Computer-Integrated Manufacturing*, vol. 93, p. 102 937, 2025. DOI: 10.1016/j.rcim.2024.102937.

[178] X. Yang, F. Amtsberg, B. Kaiser, *et al.*, "An implementation and evaluation of large-scale multi-user human–robot collaboration with head-mounted augmented reality," *Advanced Engineering Informatics*, vol. 67, p. 103 475, 2025. DOI: 10.1016/j.aei.2025.103475.

[179] M. V. A. R. Bahubalendruni and V. P. Varupala, "Disassembly sequence planning for safe disposal of end-of-life waste electric and electronic equipment," *National Academy Science Letters*, vol. 44, pp. 243–247, 2021. DOI: 10.1007/s40009-020-00994-0.

[180] G. Anil Kumar, M. V. A. R. Bahubalendruni, V. S. S. Prasad, and K. Sankaranarayanasamy, "A multi-layered disassembly sequence planning method to support decision making in de-manufacturing," *Sādhanā*, vol. 46, p. 102, 2021. DOI: 10.1007/s12046-021-01622-3.

[181] B. M. Gunji, S. K. Pabba, I. R. S. Rajaram, *et al.*, "Optimal disassembly sequence generation and disposal of parts using stability graph cut-set method for end of life product," *Sādhanā*, vol. 46, no. 1, p. 21, 2021. DOI: 10.1007/s12046-020-01525-9.

[182] A. K. Gulivindala, M. V. A. R. Bahubalendruni, M. B. P, and M. Eswaran, "Mechanical disassembly sequence planning for end-of-life products to maximize recyclability," *Sādhanā*, vol. 48, p. 109, 2023. DOI: 10.1007/s12046-023-02160-w.

[183] M. Eswaran and M. V. A. Raju Bahubalendruni, "AR/VR assisted integrated framework of autonomous disassembly system for industrial products," *Computers & Industrial Engineering*, vol. 196, p. 110 522, 2024. DOI: 10.1016/j.cie.2024.110522.

[184] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*, 2nd. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[185] E. J. Williams, "Experimental designs balanced for the estimation of residual effects of treatments," *Australian Journal of Scientific Research Series A: Physical Sciences*, vol. 2, no. 2, pp. 149–168, 1949. DOI: 10.1071/CH9490149.

[186] J. V. Bradley, "Complete counterbalancing of immediate sequential effects in a latin square design," *Journal of the American Statistical Association*, vol. 53, no. 282, pp. 525–528, 1958. DOI: 10.1080/01621459.1958.10501456.

[187] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Advances in psychology*, vol. 52, Elsevier, 1988, pp. 139–183. DOI: 10.1016/S0166-4115(08)62386-9.

[188] S. G. Hart, "NASA-task load index (NASA-TLX); 20 years later," in *Proc. Human Factors and Ergonomics Society Annual Meeting*, vol. 50, 2006, pp. 904–908. DOI: 10.1177/154193120605000909.

[189] J. Brooke *et al.*, "Sus-a quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 194, pp. 4–7, 1996. DOI: 10.1201/9781498710411-35.

[190] R. M. Taylor, "Situational awareness rating technique (SART): The development of a tool for aircrew systems design," in *Situational Awareness in Aerospace Operations (AGARD-CP-478)*, Neuilly-sur-Seine, France: NATO-AGARD, 1990, pp. 3/1–3/17.

[191] C. Bartneck, D. Kulić, E. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International Journal of Social Robotics*, vol. 1, no. 1, pp. 71–81, 2009. DOI: 10.1007/s12369-008-0001-3.

[192] K. E. Schaefer, "Measuring trust in human robot interactions: Development of the "trust perception scale-HRI"," in *Robust Intelligence and Trust in Autonomous Systems*, Boston, MA: Springer, 2016, pp. 191–218. DOI: 10.1007/978-1-4899-7668-0_10.

[193] L. J. Cronbach, "Coefficient alpha and the internal structure of tests," *Psychometrika*, vol. 16, no. 3, pp. 297–334, 1951. DOI: 10.1007/BF02310555.

[194] B. Jones and M. G. Kenward, *Design and Analysis of Cross-Over Trials*, 3rd. Boca Raton, FL: CRC Press, 2014.

[195] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *Journal of Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015. DOI: 10.18637/jss.v067.i01.

[196] A. Kuznetsova, P. B. Brockhoff, and R. H. B. Christensen, "lmerTest package: Tests in linear mixed effects models," *Journal of Statistical Software*, vol. 82, no. 13, pp. 1–26, 2017. DOI: 10.18637/jss.v082.i13.

[197] G. G. Koch, "The use of non-parametric methods in the statistical analysis of the two-period change-over design," *Biometrics*, vol. 28, no. 2, pp. 577–584, 1972. DOI: 10.2307/2556170.

[198] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

[199] A. D. Prabaswari, C. Basumerda, and B. W. Utomo, "The mental workload analysis of staff in study program of private educational organization," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 528, 2019, p. 012 018. DOI: 10.1088/1757-899X/528/1/012018.

[200] A. Bangor, P. Kortum, and J. Miller, "Determining what individual sus scores mean: Adding an adjective rating scale," *Journal of usability studies*, vol. 4, no. 3, pp. 114–123, 2009.

[201] M. Eswaran, A. K. Gulivindala, A. K. Inkulu, and M. V. A. Raju Bahubalendruni, "Augmented reality-based guidance in product assembly and maintenance/repair perspective: A state of the art review on challenges and opportunities," *Expert Systems with Applications*, vol. 213, p. 118 983, 2023. DOI: 10.1016/j.eswa.2022.118983.

[202] S. Cantarelli, D. Francia, G. M. Santi, A. Liverani, and M. Fiori, "Mixed reality lab for assembly and disassembly of industrial products," *Journal of Industrial Information Integration*, vol. 46, p. 100 873, 2025. DOI: 10.1016/j.jii.2025.100873.

Appendix A

Questionnaires

This appendix reproduces the instruments used in the operator interface study of Chapter 6: the demographics and prior experience questionnaire, the SUS, the NASA-TLX, the custom questionnaire, and the semi-structured interview guide.

A.1 Demographics Data Questionnaire

A.1.1 Demographics Questionnaire

1. Enter your given ID.
2. Enter your age.
3. What is your gender?
4. What is the highest level of education you have completed?
5. What best describes your current occupation status?
6. If you selected an employed status, please select what best represents your current role.
7. What is your field of study/professional area?
8. Do you generally use any eyesight-correcting measures?
9. If you selected “yes”, please specify.
10. Did you use your eyesight-correcting measure during the experiment?

A.1.2 Prior Experiences Questionnaire

Rated on a 5-point scale: poor, below average, average, above average, excellent.

1. How would you rate your experience with AR/VR tools or applications?

2. How would you rate your experience with assembly/disassembly activities in industrial scenarios?
3. How would you rate your experience working with robots (industrial or mobile)?

A.2 System Usability Scale Questionnaire

Items rated on a 5-point scale (1: strongly disagree; 5: strongly agree) and scored following Brooke *et al.* [189].

1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

A.3 NASA-Task Load Index Questionnaire

Items rated on an 11-point scale from 0 (very low) to 10 (very high), except item 4 (performance), which was presented with the anchors 0 (failure) and 10 (perfect) and recoded $(10 - x)$ before analysis so that lower values indicate better perceived performance, as in the original instrument. Item 7 was added following Lunding, Lystbæk, Feuchtner, *et al.* [26].

1. Mental Demand: How mentally demanding was the task?
2. Physical Demand: How physically demanding was the task?
3. Temporal Demand: How hurried or rushed was the pace of the task?
4. Performance: How successful were you in accomplishing what you were asked to do?
5. Effort: How hard did you have to work to accomplish your level of performance?
6. Frustration: How insecure, discouraged, irritated, stressed, and annoyed were you?
7. Eye Fatigue: How tiring was the task for your eyes?

A.4 Custom Questionnaire

All items were adapted from the post-trial questionnaire of Lunding, Lystbæk, Feuchtner, *et al.* [26] and rated on a 5-point Likert scale (1: strongly disagree; 5: strongly agree). Items marked with † are negatively worded and were reverse-coded before analysis ($x' = 6 - x$).

A.4.1 General Questions

1. I felt in charge of the overall disassembly task.
2. I was always aware of the overall progress during the disassembly task.
3. I performed well in the overall disassembly task.
4. It was easy to understand and follow the instructions for disassembling the car alternator.

A.4.2 Working with the Robot

1. I was always aware of the robot's next action.
2. I sometimes lost track of what the robot was doing.†
3. The robot influenced my working pace.
4. I paid more attention to the robot than to the disassembly task itself.†
5. I felt like I was working together (collaborating) with the robot.
6. I felt like the robot and I were each working independently.†

A.4.3 Augmented Reality Visualization (AR Condition Only)

Administered only after the session with the AR interface.

1. The visualizations in AR helped me understand the robot's status.
2. It was useful to preview the robot's movement overlaid on the real world before the task execution.
3. The visualization of the robot's movement made the robot's intentions (what it would do next) clear.
4. The AR instructions for disassembling the car alternator were ambiguous or unclear.†

A.5 Interview Questions

Semi-structured interview guide (approximately ten minutes, after the second session). Topics follow the interview used by Lunding, Lystbæk, Feuchtner, *et al.* [26]; follow-up questions were asked when new topics emerged.

1. Considering the interface on the touch screen, were the instructions for the disassembly task useful and clear? What did you like about them and what could be improved? Were they distracting?

2. Considering the interface on the AR glasses, were the instructions for the disassembly task (shown as holograms) useful and clear? What did you like about them and what could be improved? Were they distracting?

3. How did your experience with the touch screen differ from the AR interface? Which one did you prefer? Were you always aware of what the robot was going to do? Were you ever uncertain about its status? Were the disassembly instructions clearer in AR or in the touch screen interface?

4. Did you pay attention to the visualization of the robot's planned movements? When did you look at the movement preview and why (what did you use it for)? Was it useful? Could it have been improved in some way?

5. How would you describe your collaboration with the robot? Was the robot helpful? Could you describe it in one or two words (e.g., stressful, interesting, boring, fun, complicated, efficient, slow)?

6. Please describe how you feel about the experience. Did you find the system helpful? Would you use it frequently? Was it easy to use, or was it too complex?

Appendix B

Detailed Statistical Results of the Operator Interface Study

This appendix reports the detailed statistical results referred to in Chapter 6, which correspond to the Supplementary Material of the published article. The instruments are reproduced in Appendix A. Effect sizes are reported as Cohen's d_z (mean paired difference divided by the standard deviation of the differences). Holm-adjusted p-values (p_{Holm}) are computed within each family of secondary tests, as described in Section 6.7.3.2; exploratory subgroup and sequence-specific analyses are uncorrected unless stated otherwise.

B.1 Full Crossover Model Results and Internal Consistency

Table B.1: Primary outcomes: full crossover mixed-model results and nonparametric crossover tests. Estimates are AR minus touch screen (interface), second minus first session (session), and AR-first minus touch-screen-first sequence (sequence).

Outcome	Interface [95% CI]	p	Session [95% CI]	p	Sequence [95% CI]	p	p_{np} interface	p_{np} session	p_{np} sequence
Working with the Robot	0.25 [0.09, 0.41]	0.003	0.07 [−0.08, 0.23]	0.347	0.09 [−0.18, 0.37]	0.495	0.009	0.366	0.495
SUS	−8.68 [−13.47, −3.89]	<0.001	1.18 [−3.61, 5.97]	0.620	−6.46 [−13.96, 1.04]	0.089	<0.001	0.812	0.096
NASA-TLX (raw)	0.29 [−0.08, 0.65]	0.122	−0.49 [−0.86, −0.12]	0.010	−0.25 [−1.06, 0.56]	0.532	0.254	0.018	0.812
Eye fatigue	1.50 [0.87, 2.13]	<0.001	0.22 [−0.41, 0.85]	0.478	−1.33 [−2.55, −0.12]	0.033	<0.001	0.795	0.036
General Questions	0.17 [−0.09, 0.42]	0.188	0.04 [−0.21, 0.29]	0.739	−0.11 [−0.43, 0.21]	0.484	0.163	0.642	0.703
Total execution time (s)	49.3 [26.5, 72.0]	<0.001	−51.2 [−73.9, −28.4]	<0.001	3.6 [−37.4, 44.6]	0.860	<0.001	<0.001	0.862
Total procedure time (s)	16.7 [−22.6, 56.1]	0.393	−119.5 [−158.9, −80.2]	<0.001	−33.8 [−90.4, 22.9]	0.234	0.580	<0.001	0.248
Errors	0.14 [−0.34, 0.62]	0.560	−0.47 [−0.95, 0.01]	0.054	0.58 [0.10, 1.07]	0.020	0.138	0.006	0.008
NASA-TLX incl. eye fatigue[†]	0.46 [0.13, 0.79]	0.007	−0.39 [−0.72, −0.06]	0.021	−0.40 [−1.18, 0.38]	0.299	0.020	0.041	0.438

Linear mixed model with participant random intercept, REML, Satterthwaite df = 34. p_{np}: nonparametric crossover tests [197] (Wilcoxon rank-sum between sequences on period differences, interface differences, and participant totals).
[†]Secondary composite, not included in the Holm family.

Table B.2: Internal consistency of the questionnaire blocks (Cronbach's α).

Instrument / block	Items	Touch screen	AR
SUS	10	0.80	0.80
NASA-TLX (raw, six subscales)	6	0.72	0.70
General Questions	4	0.69	0.59
Working with the Robot	6	0.48	−0.01
AR Visualization	4	–	0.36

B.2 Exploratory Subgroup Screening

Kruskal-Wallis tests comparing participant subgroups defined by prior experience (and, for eye fatigue, by the use of visual correction), separately for each interface.

Table B.3: Statistical comparison of scores from the custom questionnaire section "Working with the Robot" across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces. Reported p-values indicate whether prior experience significantly influences participants' perceptions of collaboration and interaction with the robot.

Experience Type	Interface	p-value
AR/VR	AR glasses	0.287
AR/VR	Touch Screen	0.875
Disassembly	AR glasses	0.122
Disassembly	Touch Screen	0.221
Robot	AR glasses	0.295
Robot	Touch Screen	0.485

Table B.4: Statistical comparison of Custom Questionnaire scores for the AR Visualization section across different prior experience groups (AR/VR, Disassembly, and Robot). Reported p-values indicate whether differences between groups are statistically significant.

Experience Type	p-value
AR/VR	0.834
Disassembly	0.442
Robot	0.534

Table B.5: Statistical comparison of eye-fatigue scores between participants who did and did not use visual correction, showing Kruskal-Wallis p-values for each interface.

Interface	p_value
AR glasses	0.284
Touch Screen	0.253

Table B.6: Statistical comparison of NASA-TLX scores across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces, including p-values for participants with and without eye-fatigue.

Experience	**Interface**	**With eye-fatigue**	**Without eye-fatigue**
AR/VR	AR glasses	0.538	0.639
	Touch Screen	0.483	0.509
Disassembly	AR glasses	0.568	0.695
	Touch Screen	0.0720	0.0398
Robot	AR glasses	0.706	0.529
	Touch Screen	0.733	0.771

Table B.7: Statistical comparison of SUS scores across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces. Reported p-values indicate whether prior experience significantly influences perceived usability across interaction modalities.

Experience Type	Interface	p-value
AR/VR	AR glasses	0.689
AR/VR	Touch Screen	0.687
Disassembly	AR glasses	0.212
Disassembly	Touch Screen	0.113
Robot	AR glasses	0.245
Robot	Touch Screen	0.574

Table B.8: Statistical comparison of scores from the custom questionnaire general questions across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces. Reported p-values indicate whether prior experience significantly influences participants' subjective evaluations.

Experience Type	Interface	p-value
AR/VR	AR glasses	0.202
AR/VR	Touch Screen	0.428
Disassembly	AR glasses	0.685
Disassembly	Touch Screen	0.245
Robot	AR glasses	0.0654
Robot	Touch Screen	0.0962

Table B.9: Statistical comparison of error rates across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces. Reported p-values indicate whether differences between groups are statistically significant.

Experience Type	Interface	p-value
AR/VR	AR glasses	0.190
AR/VR	Touch Screen	0.247
Disassembly	AR glasses	0.528
Disassembly	Touch Screen	0.378
Robot	AR glasses	0.653
Robot	Touch Screen	0.673

Table B.10: Comparison of execution and total procedure times across different prior experience groups (AR/VR, Disassembly, and Robot) for both AR glasses and Touch Screen interfaces. The table reports p-values for each metric type, highlighting statistically significant differences in task duration related to prior experience and interface type.

Experience Type	Interface	Metric Type	p-value
AR/VR	AR glasses	Execution Time	0.408
AR/VR	AR glasses	Total Procedure Time	0.317
AR/VR	Touch Screen	Execution Time	0.793
AR/VR	Touch Screen	Total Procedure Time	0.595
Disassembly	AR glasses	Execution Time	0.249
Disassembly	AR glasses	Total Procedure Time	0.251
Disassembly	Touch Screen	Execution Time	0.225
Disassembly	Touch Screen	Total Procedure Time	0.251
Robot	AR glasses	Execution Time	0.495
Robot	AR glasses	Total Procedure Time	0.113
Robot	Touch Screen	Execution Time	0.647
Robot	Touch Screen	Total Procedure Time	0.667

B.3 Exploratory Subgroup Follow-up

B.3.1 Prior Disassembly Experience and Workload

The raw NASA-TLX score with the touch screen differed between disassembly-experience groups ($p = 0.040$, Table B.6). Within-group paired comparisons are reported in Table B.11 and Table B.12, and the corresponding means in Fig. B.1 and Fig. B.2.

Table B.11: Descriptive statistics of NASA-TLX scores categorized by participants' disassembly experience levels (Above-Average, Average, Below-Average, Excellent, Poor) for both conditions with and without eye-fatigue. The table includes the statistical test used, p-values, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals for each subgroup.

Score Type	Disassembly Experience	Test Used	p_value	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
With eye-fatigue	Above-Average	Paired t-test	0.430	0.926	0.537	11	0.248	0.95	-0.358	0.843
With eye-fatigue	Average	Paired t-test	0.302	0.644	0.560	8	0.394	0.95	-0.341	1.10
With eye-fatigue	Below-Average	Paired t-test	0.440	0.222	0.111	8	0.290	0.95	-0.429	0.989
With eye-fatigue	Excellent	Wilcoxon signed-rank	0.371	NA	NA	2	0.990	0.95	-0.908	2.73
With eye-fatigue	Poor	Paired t-test	0.0346	0.0771	NA	7	0.933	0.95	0.00570	1.81
Without eye-fatigue	Above-Average	Paired t-test	0.780	0.747	0.286	11	0.0867	0.95	-0.508	0.677
Without eye-fatigue	Average	Paired t-test	0.521	0.889	0.895	8	0.239	0.95	-0.473	0.935
Without eye-fatigue	Below-Average	Paired t-test	0.703	0.226	0.0643	8	-0.140	0.95	-0.832	0.561
Without eye-fatigue	Excellent	Wilcoxon signed-rank	0.371	NA	NA	2	0.990	0.95	-0.908	2.73
Without eye-fatigue	Poor	Paired t-test	0.205	0.153	NA	7	0.703	0.95	-0.156	1.52

B.3.2 Prior Robot Experience and General Task Experience

The General Questions index showed borderline differences between robot-experience groups (Table B.8). Within-group paired comparisons are reported in Table B.13 and Table B.14, and the corresponding means in Fig. B.4 and Fig. B.3.

Table B.12: Descriptive statistics of NASA-TLX components analyzed by participants' disassembly experience levels. For each component and experience subgroup, the table reports the statistical test used, p-value, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals.

Component	Disassembly Experience	Test Used	p_value	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
Effort	Above-Average	Paired t-test	0.567	0.0986	0.152	11	-0.179	0.95	-0.770	0.422
Effort	Average	Paired t-test	0.502	0.522	0.387	8	0.250	0.95	-0.463	0.946
Effort	Below-Average	Paired t-test	0.685	0.155	0.481	8	0.150	0.95	-0.553	0.842
Effort	Excellent	Wilcoxon signed-rank	1	NA	NA	2	0.707	0.95	-1.00	2.24
Effort	Poor	Wilcoxon signed-rank	0.203	0.549	NA	7	0.586	0.95	-0.243	1.37
eye-fatigue	Above-Average	Paired t-test	0.0569	0.289	0.385	11	0.649	0.95	-0.0183	1.29
eye-fatigue	Average	Wilcoxon signed-rank	0.0975	0.0130	0.0328	8	0.713	0.95	-0.0899	1.48
eye-fatigue	Below-Average	Paired t-test	0.00444	0.516	0.907	8	1.46	0.95	0.416	2.45
eye-fatigue	Excellent	No variation	NA	NA	NA	2	0	NA	NA	NA
eye-fatigue	Poor	Wilcoxon signed-rank	0.0899	0.958	NA	7	0.831	0.95	-0.0643	1.68
Frustration	Above-Average	Wilcoxon signed-rank	0.289	0.0251	0.954	11	0.252	0.95	-0.355	0.847
Frustration	Average	Paired t-test	0.550	0.244	0.237	8	0.222	0.95	-0.488	0.917
Frustration	Below-Average	Wilcoxon signed-rank	1	0.00981	0.0563	8	0	0.95	-0.693	0.693
Frustration	Excellent	No variation	NA	NA	NA	2	0	NA	NA	NA
Frustration	Poor	Wilcoxon signed-rank	0.0975	0.0217	NA	7	0.731	0.95	-0.135	1.55
Mental	Above-Average	Wilcoxon signed-rank	1	0.0338	0.179	11	-0.0535	0.95	-0.644	0.539
Mental	Average	Paired t-test	0.844	0.196	0.578	8	0.0724	0.95	-0.624	0.764
Mental	Below-Average	Wilcoxon signed-rank	0.784	0.0370	0.118	8	-0.168	0.95	-0.861	0.536
Mental	Excellent	Wilcoxon signed-rank	0.371	NA	NA	2	1.18	0.95	-0.855	3.08
Mental	Poor	Wilcoxon signed-rank	0.136	0.686	NA	7	0.690	0.95	-0.165	1.50
Performance	Above-Average	Wilcoxon signed-rank	0.589	0.0317	0.252	11	-0.169	0.95	-0.760	0.431
Performance	Average	Wilcoxon signed-rank	0.203	0.0114	0.00962	8	-0.484	0.95	-1.21	0.267
Performance	Below-Average	Wilcoxon signed-rank	1	0.00167	0.00748	8	0.100	0.95	-0.598	0.792
Performance	Excellent	Wilcoxon signed-rank	0.371	NA	NA	2	2.12	0.95	-0.654	5.01
Performance	Poor	Wilcoxon signed-rank	0.346	0.000275	NA	7	0.586	0.95	-0.243	1.37
Physical	Above-Average	Wilcoxon signed-rank	0.223	0.00825	0.0205	11	0.442	0.95	-0.189	1.05
Physical	Average	Wilcoxon signed-rank	1	0.000168	0.00135	8	0.195	0.95	-0.512	0.889
Physical	Below-Average	Wilcoxon signed-rank	1	0.00000105	0.000355	8	-0.354	0.95	-1.06	0.374
Physical	Excellent	Wilcoxon signed-rank	0.371	NA	NA	2	1.41	0.95	-0.796	3.54
Physical	Poor	Wilcoxon signed-rank	0.174	0.0201	NA	7	0.726	0.95	-0.139	1.55
Temporal	Above-Average	Paired t-test	0.736	0.0754	0.341	11	0.104	0.95	-0.491	0.695
Temporal	Average	Paired t-test	0.250	0.557	0.518	8	-0.444	0.95	-1.16	0.300
Temporal	Below-Average	Paired t-test	0.442	0.178	0.578	8	-0.288	0.95	-0.987	0.430
Temporal	Excellent	No Variation	NA	NA	NA	2	0	NA	NA	NA
Temporal	Poor	Wilcoxon signed-rank	0.854	0.0847	NA	7	-0.109	0.95	-0.848	0.639

Table B.13: Descriptive statistics for the custom questionnaire general questions grouped by participants' prior experience working with robots (Above-Average, Average, Below-Average, Excellent, Poor). The table reports the statistical test used, p-values, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals for each subgroup, highlighting potential differences in subjective responses based on prior robot interaction experience. Entries with NA indicate insufficient data for statistical analysis.

Robot Experience	Test Used	p_value	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
Above-Average	Paired t-test	0.0915	0.0738	0.0798	20	0.397	0.95	-0.0632	0.849
Average	Wilcoxon signed-rank	0.586	0.850	NA	4	0.439	0.95	-0.627	1.44
Below-Average	Wilcoxon signed-rank	1	0.363	NA	3	-0.378	0.95	-1.52	0.843
Excellent	Paired t-test	0.291	0.107	0.138	8	0.404	0.95	-0.332	1.11
Poor	NA	NA	NA	NA	1	NA	NA	NA	NA

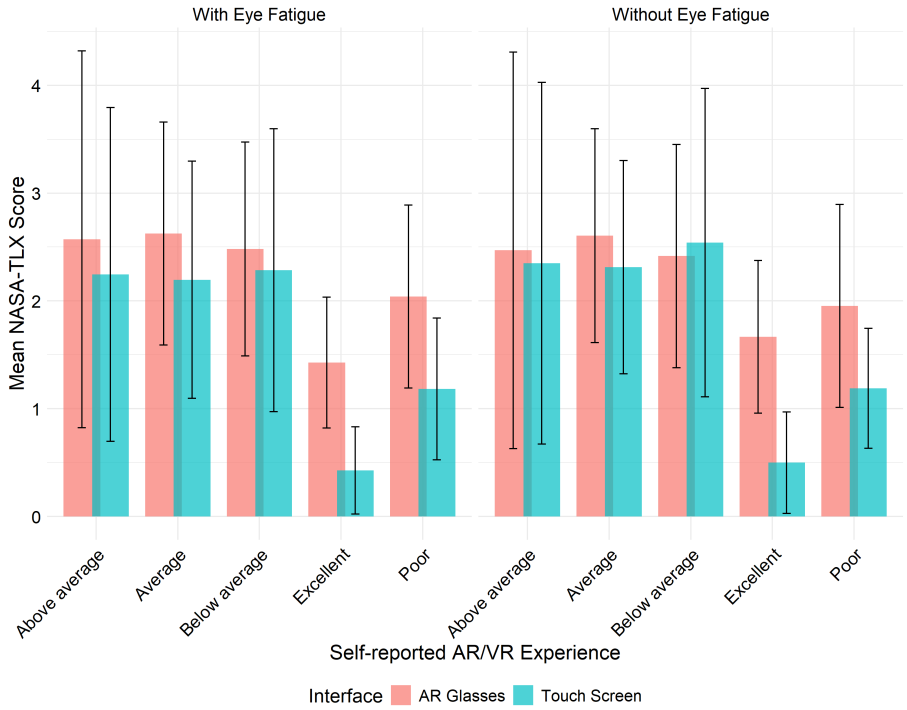


Figure B.1: Mean NASA-TLX workload scores grouped by participants' self-reported disassembly experience, highlighting how prior expertise influences perceived task load levels.

Table B.14: Descriptive statistics for individual components of the custom general questions questionnaire (items 1-4) grouped by participants' prior experience working with robots (Above-Average, Average, Below-Average, Excellent, Poor). For each component and subgroup, the table reports the statistical test used, p-values, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals. Refer to Appendix A.4.1 for the corresponding question definitions.

Component	Robot Experience	Test Used	p_value	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
1	Above-Average	Wilcoxon signed-rank	0.618	0.000555	0.673	20	0.117	0.95	-0.324	0.556
1	Average	Wilcoxon signed-rank	0.346	0.0239	NA	4	0.866	0.95	-0.359	2.00
1	Below-Average	Wilcoxon signed-rank	0.371	0.463	NA	3	-0.801	0.95	-2.09	0.603
1	Excellent	Wilcoxon signed-rank	0.0719	0.000892	1	8	0.935	0.95	0.0706	1.76
1	Poor	NA	NA	NA	NA	1	NA	NA	NA	NA
2	Above-Average	Wilcoxon signed-rank	0.148	0.0446	0.294	20	0.358	0.95	-0.0987	0.807
2	Average	Wilcoxon signed-rank	1	0.00124	NA	4	0.5	0.95	-0.585	1.52
2	Below-Average	Wilcoxon signed-rank	1	0	NA	3	0.577	0.95	-0.721	1.77
2	Excellent	Wilcoxon signed-rank	1	0.00120	0.0261	8	0.150	0.95	-0.553	0.842
2	Poor	NA	NA	NA	NA	1	NA	NA	NA	NA
3	Above-Average	Wilcoxon signed-rank	0.240	0.000619	0.136	20	0.287	0.95	-0.164	0.731
3	Average	Wilcoxon signed-rank	1	0.683	NA	4	0	0.95	-0.980	0.980
3	Below-Average	Wilcoxon signed-rank	1	0	NA	3	0.577	0.95	-0.721	1.77
3	Excellent	Wilcoxon signed-rank	1	0.00518	1	8	0	0.95	-0.693	0.693
3	Poor	NA	NA	NA	NA	1	NA	NA	NA	NA
4	Above-Average	Paired t-test	0.134	0.109	0.573	20	0.350	0.95	-0.106	0.798
4	Average	No variation	NA	NA	NA	4	0	NA	NA	NA
4	Below-Average	No variation	NA	NA	NA	3	0	NA	NA	NA
4	Excellent	Paired t-test	0.451	0.0540	0.170	8	0.282	0.95	-0.435	0.981
4	Poor	NA	NA	NA	NA	1	NA	NA	NA	NA

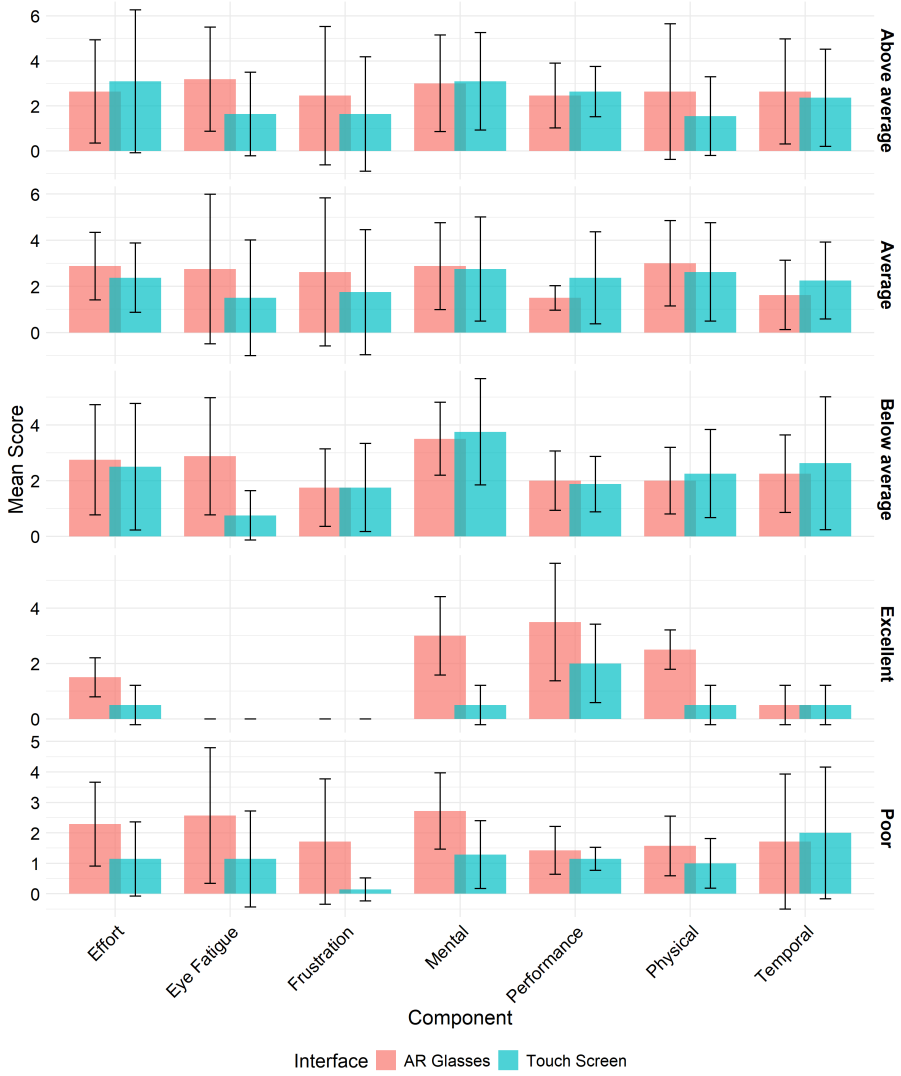


Figure B.2: Mean NASA-TLX component scores (mental demand, physical demand, temporal demand, effort, performance, frustration) grouped by participants' self-reported disassembly experience, illustrating how prior experience affects specific dimensions of perceived workload.

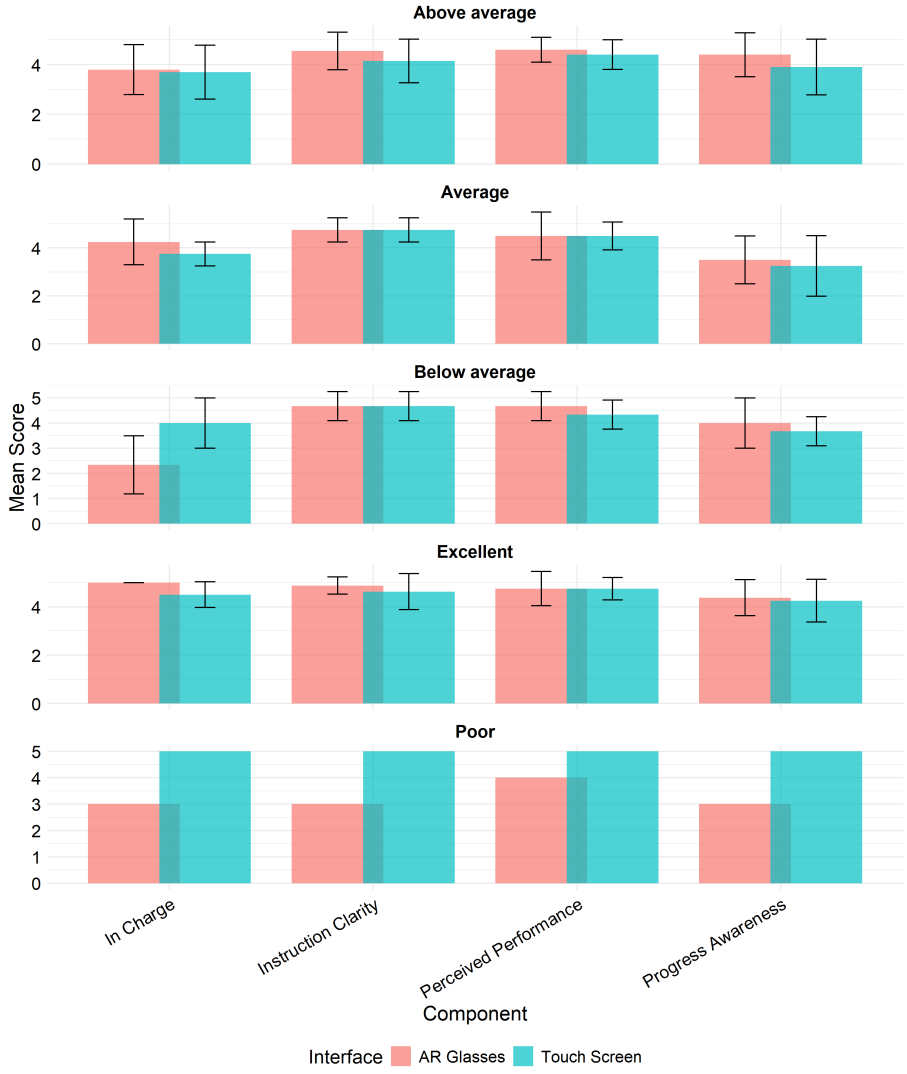


Figure B.3: Mean scores of each component of the general questions from the custom questionnaire, grouped by participants' prior experience with robots. This breakdown illustrates how different aspects of user perception vary according to robot experience.

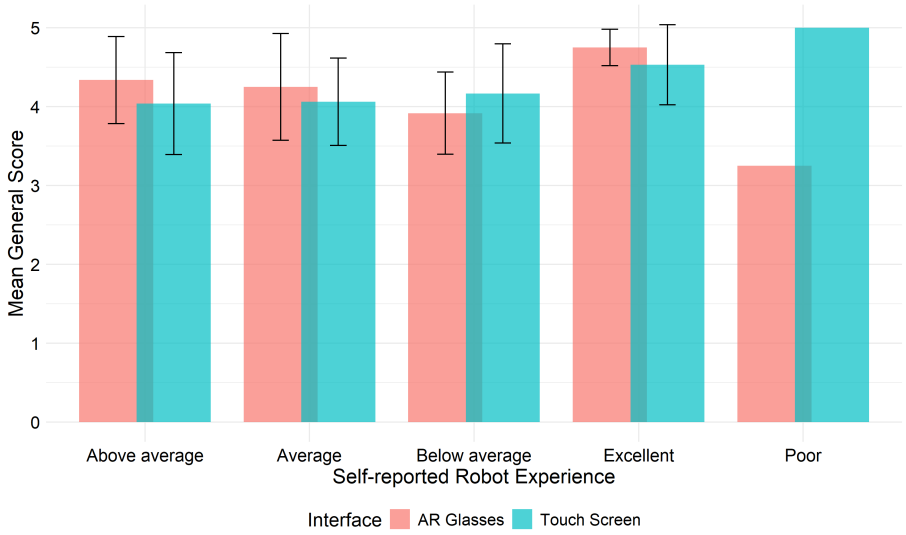


Figure B.4: Mean scores of the general questions from the custom questionnaire, grouped by participants' prior experience with robots, highlighting how robot experience influences overall perceptions and responses.

B.4 Item-, Subscale-, and Operation-Level Analyses

Table B.15: Descriptive statistics for individual NASA-TLX components across the study groups, including the statistical test used, p-values, normality tests (Shapiro and D'Agostino), sample size, effect size (Cohen's d_z), and 95% confidence intervals for each component (Effort, eye-fatigue, Frustration, Mental Demand, Performance, Physical Demand, and Temporal Demand).

Component	Test Used	p-value	p_{Holm}	Shapiro p	D'Agostino p	n	Cohen's d_z	CI	CI Low	CI High
Effort	Wilcoxon signed-rank	0.218	0.872	0.0220	0.0489	36	0.148	0.95	-0.182	0.475
Eye-fatigue	Wilcoxon signed-rank	0.000122	<0.001	0.0120	0.125	36	0.813	0.95	0.431	1.19
Frustration	Wilcoxon signed-rank	0.0506	0.304	0.0000239	0.697	36	0.276	0.95	-0.0590	0.607
Mental	Paired t-test	0.247	0.872	0.0951	0.609	36	0.196	0.95	-0.135	0.525
Performance	Wilcoxon signed-rank	0.855	1.000	0.0000772	0.00363	36	-0.0643	0.95	-0.391	0.263
Physical	Wilcoxon signed-rank	0.0755	0.378	0.000000675	0.0000493	36	0.331	0.95	-0.00697	0.664
Temporal	Wilcoxon signed-rank	0.648	1.000	0.0211	0.382	36	-0.0973	0.95	-0.424	0.231

p_{Holm}: Holm-adjusted across the seven NASA-TLX items (six subscales and eye fatigue).

Table B.16: Descriptive statistics for individual SUS questionnaire components (items 1-10) comparing interfaces in the group analysis. The table reports the statistical test used, p-values, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals for each item. Refer to Appendix A.2 for the corresponding question definitions.

Component	Test Used	p_value	p_{Holm}	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
1	Wilcoxon signed-rank	0.00970	0.068	0.00454	0.445	36	-0.462	0.95	-0.803	-0.115
2	Wilcoxon signed-rank	0.196	0.800	0.000434	0.0956	36	0.225	0.95	-0.108	0.554
3	Wilcoxon signed-rank	0.0148	0.089	0.00480	0.109	36	-0.441	0.95	-0.781	-0.0956
4	Wilcoxon signed-rank	0.00388	0.039	0.00000649	0.00104	36	0.527	0.95	0.175	0.873
5	Wilcoxon signed-rank	0.342	1.000	0.000826	0.101	36	-0.172	0.95	-0.500	0.159
6	Wilcoxon signed-rank	0.160	0.800	0.00000305	0.00821	36	0.251	0.95	-0.0827	0.581
7	Wilcoxon signed-rank	0.00426	0.039	0.000981	0.158	36	-0.527	0.95	-0.873	-0.175
8	Wilcoxon signed-rank	0.00841	0.067	0.0195	0.539	36	0.491	0.95	0.141	0.834
9	Wilcoxon signed-rank	0.487	1.000	0.00517	0.680	36	-0.113	0.95	-0.440	0.215
10	Wilcoxon signed-rank	0.685	1.000	0.0000364	0.478	36	0.0703	0.95	-0.257	0.397

p_{Holm}: Holm-adjusted across the ten SUS items.

Table B.17: Descriptive statistics for individual components of the custom general questions questionnaire (items 1-4) comparing interfaces in the group analysis. The table reports the statistical test used, p-values, normality test results (Shapiro and D'Agostino), sample size (n), effect size (Cohen's d_z), and 95% confidence intervals for each component. Refer to Appendix A.4.1 for the corresponding question definitions.

Component	Test Used	p_value	p_{Holm}	Shapiro_p	D'Agostino_p	n	Cohen's d_z	CI	CI_low	CI_high
1	Wilcoxon signed-rank	0.603	0.693	0.00000885	0.000412	36	0.0257	0.95	-0.301	0.352
2	Wilcoxon signed-rank	0.159	0.636	0.000388	0.0943	36	0.257	0.95	-0.0773	0.587
3	Wilcoxon signed-rank	0.336	0.693	0.0000121	0.203	36	0.167	0.95	-0.163	0.494
4	Wilcoxon signed-rank	0.231	0.693	0.000228	0.307	36	0.218	0.95	-0.114	0.548

p_{Holm}: Holm-adjusted across the four General Questions items.

Table B.18: Descriptive statistics for execution times for each task (O0-O8), including the statistical test, p-values, normality assessments, skewness, kurtosis, and effect sizes. This analysis highlights task-level differences in execution speed between the AR glasses and Touch Screen interfaces.

Task	Test Used	p-value	p_{Holm}	Statistic	Shapiro p	D'Agostino p	Skewness	Kurtosis	n	Cohen's d_z	CI	CI_{low}	CI_{high}
O0	Wilcoxon signed-rank	0.000190	0.001	516	1.01×10^{-5}	6.30×10^{-6}	2.28	10.1	36	0.597	0.95	0.239	0.949
O1	Wilcoxon signed-rank	0.0000160	<0.001	522	6.50×10^{-4}	2.14×10^{-4}	1.67	8.17	36	0.815	0.95	0.433	1.19
O2	Wilcoxon signed-rank	3.30×10^{-7}	<0.001	626	1.30×10^{-8}	8.41×10^{-8}	3.09	13.1	36	0.742	0.95	0.368	1.11
O3	Paired t-test	0.305	0.305	1.04	0.257	0.614	0.180	3.51	36	0.174	0.95	-0.157	0.502
O4	Wilcoxon signed-rank	0.00552	0.017	510	4.99×10^{-6}	2.02×10^{-6}	2.49	11.5	36	0.486	0.95	0.137	0.828
O5	Wilcoxon signed-rank	0.000129	<0.001	418	3.09×10^{-10}	6.79×10^{-10}	4.10	20.5	36	0.403	0.95	0.0601	0.740
O6	Wilcoxon signed-rank	0.00379	0.015	394	1.28×10^{-3}	8.41×10^{-2}	0.648	6.58	36	0.426	0.95	0.0815	0.764
O7	Wilcoxon signed-rank	0.0788	0.158	338	1.24×10^{-2}	5.10×10^{-3}	1.15	5.99	36	0.329	0.95	-0.00841	0.663
O8	Wilcoxon signed-rank	0.00159	0.008	482	1.58×10^{-3}	2.47×10^{-3}	1.27	6.66	36	0.511	0.95	0.160	0.855

p_{Holm}: Holm-adjusted across the nine operations.

B.5 Error Type and Location

Table B.19: Comparison of error types between AR glasses and Touch Screen interfaces using Fisher's Exact Test. The table reports the number of occurrences for each error type per interface and the corresponding p-values, highlighting significant differences in user errors between interaction modalities.

Error Type	AR_n	TS_n	p_value	p_{Holm}
Early Finish	0	5	0.0165	0.083
Forgot Continue	6	4	0.741	1.000
Forgot END	3	0	0.243	0.972
Forgot INI	21	17	1.000	1.000
Forgot Piece	3	2	1.000	1.000

p_{Holm}: Holm-adjusted across the five error types.

Table B.20: Comparison of errors by operation between AR glasses and Touch Screen interfaces using Fisher's Exact Test. The table shows the number of errors observed for each operation (O0-O8) per interface along with p-values, indicating whether error rates differ significantly between interaction modalities.

Operation	AR_n	TS_n	p_value	p_{Holm}
O0	6	5	1.000	1.000
O1	0	2	0.207	1.000
O2	0	1	0.459	1.000
O3	7	7	0.768	1.000
O4	7	6	1.000	1.000
O5	0	1	0.459	1.000
O6	0	1	0.459	1.000
O7	10	4	0.222	1.000
O8	3	1	0.618	1.000

p_{Holm}: Holm-adjusted across the nine error locations.

Table B.21: Global chi-square test assessing the distribution of errors across all operations (O0-O8) to determine whether the overall pattern of errors differs between the AR glasses and Touch Screen interfaces. The table reports the chi-square statistic, degrees of freedom, and p-value for group-level analysis.

Test	χ^2	df	p-value
Chi-square	8.39	8	0.397

B.6 Sequence-Specific Analyses

The following analyses compare the first and second session within each sequence. Because session and interface are confounded within a sequence, they combine interface and practice effects (Section 6.7.4.4.3) and are reported descriptively; the interface and session effects are estimated jointly by the crossover model (Table 6.4).

Table B.22: Descriptive statistics of total execution and procedure times broken down by initial interface (AR glasses and Touch Screen). Includes test type, p-values, normality assessments (Shapiro-Wilk and D'Agostino), skewness, kurtosis, sample size, and effect sizes (Cohen's d_z with 95% confidence intervals). This analysis highlights performance differences attributable to the first interface used.

First Interface	Metric Type	Test Used	p-value	Statistic	Shapiro p	D'Agostino p	Skewness	Kurtosis	n	Cohen's d_z	CI	CI_{low}	CI_{high}
AR glasses	Total Execution Time	Wilcoxon signed-rank	0.000214	171	0.0113	0.0384	1.04	3.26	18	1.27	0.95	0.636	1.89
AR glasses	Total Procedure Time	Paired t-test	0.0000439	5.44	0.668	0.516	0.302	2.79	18	1.28	0.95	0.643	1.90
Touch Screen	Total Execution Time	Paired t-test	0.878	-0.156	0.940	0.999	0.000772	2.84	18	-0.0368	0.95	-0.498	0.426
Touch Screen	Total Procedure Time	Paired t-test	0.00288	-3.48	0.747	0.723	-0.164	2.91	18	-0.820	0.95	-1.35	-0.274

Table B.23: Descriptive statistics of execution times for each task (O0-O8) separated by initial interface (AR glasses vs. Touch Screen). The table presents the statistical test used, p-values, normality assessments (Shapiro-Wilk and D'Agostino), skewness, kurtosis, sample size, and effect sizes (Cohen's d_z with 95% confidence intervals). This breakdown highlights interface-specific performance differences for individual operations, providing insight into task complexity and interface influence on execution speed.

Task	First Interface	Test Used	p-value	Statistic	Shapiro p	D'Agostino p	Skewness	Kurtosis	n	Cohen's d_z	CI	CI_{low}	CI_{high}
O0	AR glasses	Wilcoxon signed-rank	0.00107	161	0.000876	0.00105	1.85	6.42	18	0.758	0.95	0.222	1.28
O0	Touch Screen	Paired t-test	0.0571	2.04	0.991	0.768	-0.137	2.37	18	0.481	0.95	-0.0142	0.964
O1	AR glasses	Paired t-test	0.0000312	5.61	0.911	0.849	0.0882	2.18	18	1.32	0.95	0.674	1.95
O1	Touch Screen	Wilcoxon signed-rank	0.0132	129	0.000907	0.000414	2.06	8.02	18	0.560	0.95	0.0543	1.05
O2	AR glasses	Wilcoxon signed-rank	0.000303	153	0.0202	0.00787	1.40	5.14	18	1.32	0.95	0.675	1.95
O2	Touch Screen	Wilcoxon signed-rank	0.000324	168	1.61×10^{-5}	0.000170	2.26	7.18	18	0.674	0.95	0.152	1.18
O3	AR glasses	Paired t-test	0.000939	3.99	0.0923	0.131	0.730	2.48	18	0.941	0.95	0.373	1.49
O3	Touch Screen	Paired t-test	0.0233	-2.49	0.172	0.124	-0.745	3.21	18	-0.588	0.95	-1.08	-0.0786
O4	AR glasses	Wilcoxon signed-rank	0.00147	159	0.000366	0.000203	2.22	8.21	18	0.715	0.95	0.187	1.23
O4	Touch Screen	Paired t-test	0.412	0.842	0.103	0.214	0.591	2.19	18	0.198	0.95	-0.271	0.662
O5	AR glasses	Wilcoxon signed-rank	0.00607	108	1.29×10^{-7}	4.40×10^{-7}	3.69	15.1	18	0.361	0.95	-0.122	0.833
O5	Touch Screen	Wilcoxon signed-rank	0.0100	106	0.0000455	0.0000971	2.39	8.38	18	0.532	0.95	0.0302	1.02
O6	AR glasses	Paired t-test	0.170	1.43	0.0942	0.270	0.522	4.62	18	0.338	0.95	-0.142	0.809
O6	Touch Screen	Paired t-test	0.00915	2.94	0.246	0.227	0.574	3.24	18	0.693	0.95	0.168	1.20
O7	AR glasses	Wilcoxon signed-rank	0.157	75.5	0.0637	0.0333	1.07	5.15	18	0.367	0.95	-0.115	0.840
O7	Touch Screen	Paired t-test	0.246	1.20	0.707	0.285	0.505	2.84	18	0.283	0.95	-0.192	0.751
O8	AR glasses	Paired t-test	0.00845	2.98	0.0897	0.0842	0.845	4.60	18	0.702	0.95	0.176	1.21
O8	Touch Screen	Paired t-test	0.233	1.24	0.300	0.331	-0.457	3.61	18	0.292	0.95	-0.185	0.760

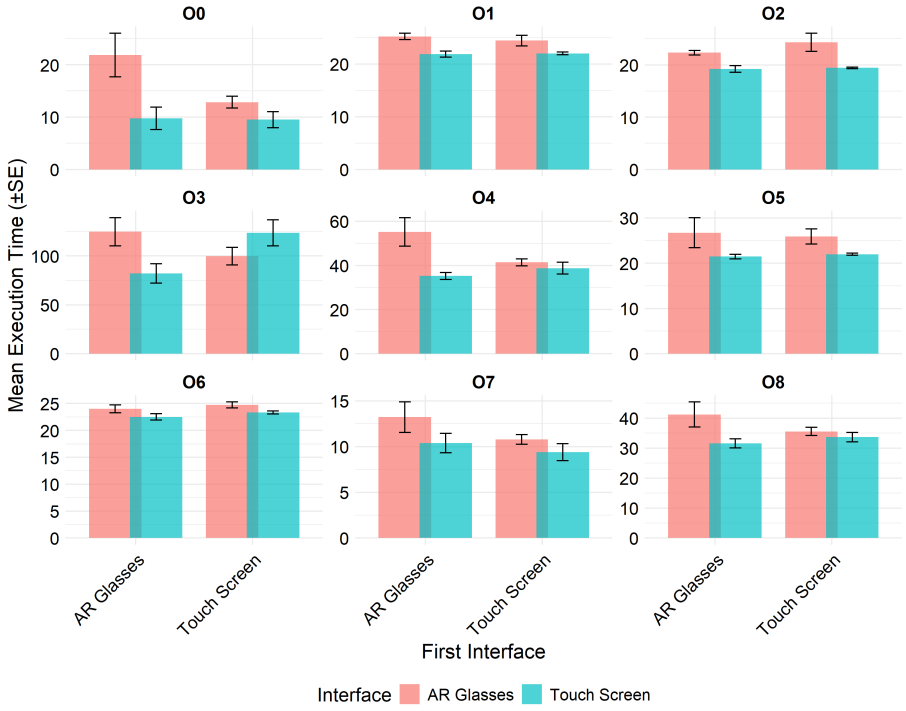


Figure B.5: Mean execution times for each operation, analyzed according to the first interface used, highlighting the effect of interface order on task performance.

Table B.24: Descriptive statistics of errors based on participants' first interface experience (AR glasses vs. Touch Screen). The table reports the statistical test used, p-value, test statistic, normality test results (Shapiro and D'Agostino), skewness, kurtosis, sample size (n), effect size (Cohen's d_z), and 95% confidence intervals. This analysis highlights differences in error patterns depending on the initial interface encountered.

First Interface	Test Used	p-value	p_{Holm}	Statistic	Shapiro p	D'Agostino p	Skewness	Kurtosis	n	Cohen's d_z	CI	CI_{low}	CI_{high}
AR glasses	Wilcoxon signed-rank	0.0441	0.088	95	0.00186	0.00130	-1.81	6.93	18	0.348	0.95	-0.164	0.861
Touch Screen	Wilcoxon signed-rank	0.183	0.183	15	0.00742	0.0723	-0.884	4.31	18	-0.344	0.95	-0.855	0.168

p_{Holm}: Holm-adjusted across the two sequences.

Table B.25: Descriptive statistics of error types based on participants' first interface experience (AR glasses vs. Touch Screen). Counts for each error type (AR_n, TS_n) and total occurrences (Total_n) are reported along with Fisher's Exact Test p-values, highlighting differences in error patterns attributable to the initial interface.

First Interface	Error Type	AR_n	TS_n	Total_n	p_value
AR glasses	Ended Task Before Robot Stop	0	5	5	0.00401
AR glasses	Forgot Continue	5	1	6	0.388
AR glasses	Forgot END	3	0	3	0.287
AR glasses	Forgot INI	16	9	25	1
AR glasses	Forgot Piece	2	0	2	0.524
Touch Screen	Forgot Continue	1	3	4	1
Touch Screen	Forgot INI	5	8	13	1
Touch Screen	Forgot Piece	1	2	3	1

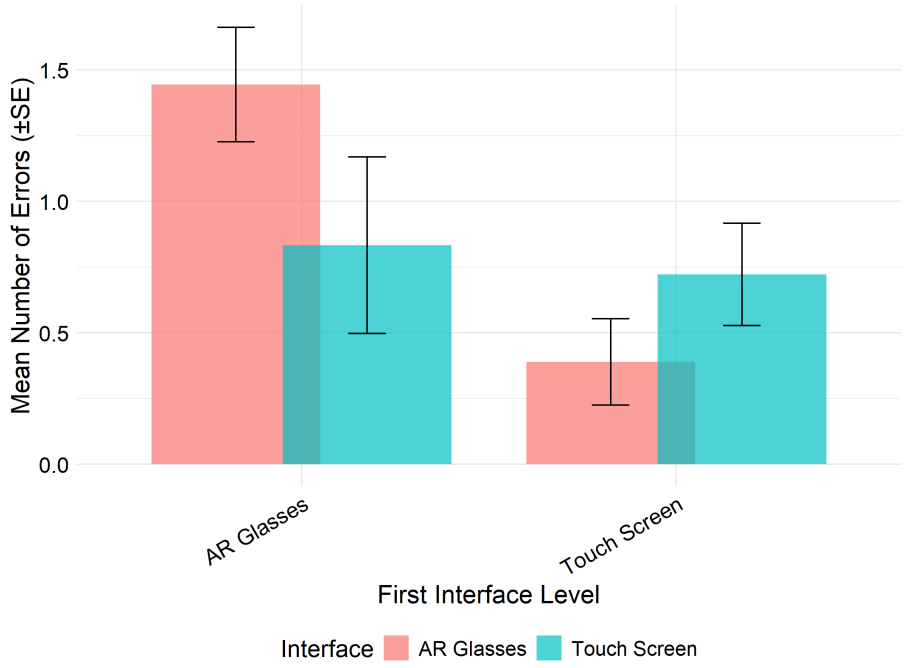


Figure B.6: Mean number of errors made by participants depending on which interface they used first, highlighting potential learning effects or initial usability challenges associated with each interface.

Table B.26: Descriptive statistics of operation-specific errors according to participants' first interface experience (AR glasses vs. Touch Screen). The table presents counts per operation (AR_n, TS_n), total occurrences (Total_n), and Fisher's Exact Test p-values, allowing comparison of error distributions across initial interfaces.

First Interface	Operation	AR_n	TS_n	Total_n	p_value
AR glasses	O0	6	3	9	1
AR glasses	O1	0	1	1	0.366
AR glasses	O2	0	1	1	0.366
AR glasses	O3	4	4	8	0.434
AR glasses	O4	6	1	7	0.232
AR glasses	O5	0	1	1	0.366
AR glasses	O6	0	1	1	0.366
AR glasses	O7	8	2	10	0.277
AR glasses	O8	2	1	3	1
Touch Screen	O0	0	2	2	0.521
Touch Screen	O1	0	1	1	1
Touch Screen	O3	3	3	6	0.613
Touch Screen	O4	1	5	6	0.354
Touch Screen	O7	2	2	4	0.587
Touch Screen	O8	1	0	1	0.350

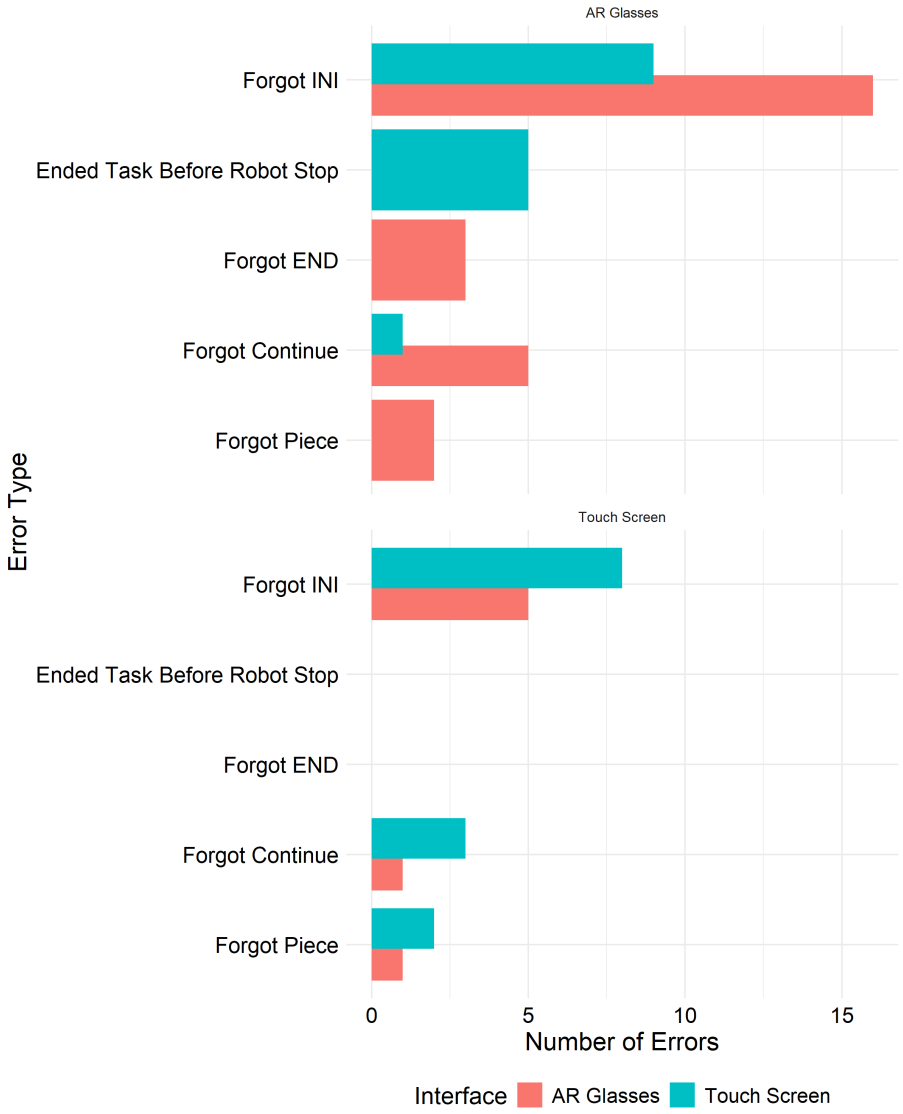


Figure B.7: Frequency of errors categorized by type, depending on which interface participants used first, highlighting how initial interface exposure may influence error patterns and learning effects.

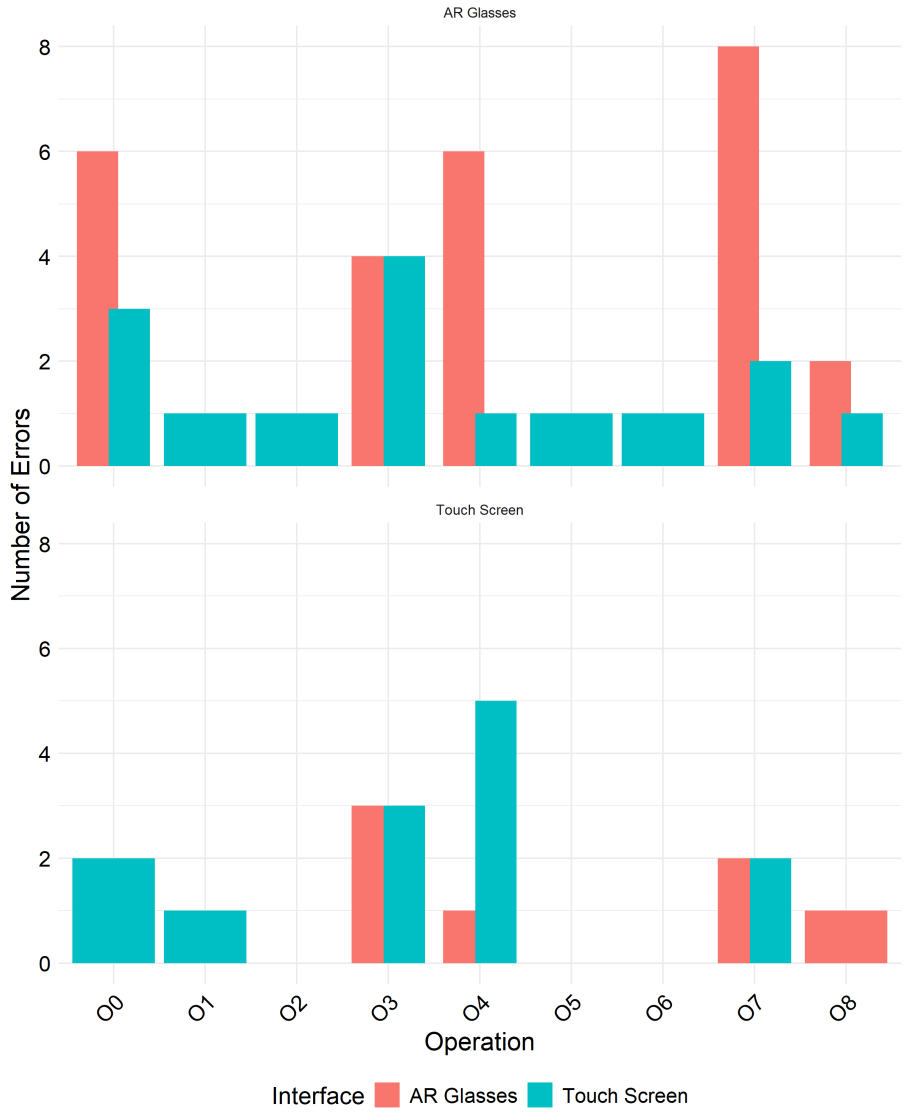


Figure B.8: Frequency of errors broken down by assembly operation and the first interface participants used, illustrating how initial interface exposure affects error occurrence across different task steps.