Predictive Object Transfer in Remanufacturing: A Cyber-Physical Approach Using Digital Twins

Zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

von der KIT-Fakultät für Maschinenbau des
Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

M.Sc. Jan-Felix Klein
geb. in Marburg

Tag der mündlichen Prüfung:	10.01.2025
Hauptreferent:	Prof. Dr.-Ing. Kai Furmans
Korreferent:	Prof. Tullio Tolio

Abstract

To mitigate the effects of global warming, governments worldwide have set targets for reducing greenhouse gas emissions. Achieving these targets necessitates a socio-ecological transformation in the industrial sector, shifting from the linear "take, make, waste" model to a circular economy. Remanufacturing, one of the end-of-life strategies, is often considered to be the most effective strategy for saving materials and energy (Nasr and Thurston 2006). However, economic barriers at both the supply chain and factory levels hinder the broader implementation of remanufacturing across various industries. The primary challenge is the high level of uncertainty associated with returned products, requiring significant process flexibility that current automation approaches cannot accommodate. Industry 4.0 methods are frequently proposed as a solution to reduce this uncertainty in remanufacturing processes. Nonetheless, there is a notable scarcity of implementations that effectively demonstrate these benefits, highlighting a critical research gap.

In this thesis, we contribute to closing this gap by demonstrating a real-world application of a flexible, digital twin-driven object transfer system. Our contributions are four-fold. First, we propose a non-hierarchical Digital Twin architecture. Second, we develop a formal description model for a generalised object transfer system. Third, we implement both the architecture and the model in a real-world cyber-physical system for object transfer within a remanufacturing demonstration environment. Finally, we conduct an application study using the proposed system, where digital twins are employed to forecast the most promising pre-transfer states for starter motor cores that have never been physically transferred.

The proposed Digital Twin architecture is characterised by a non-hierarchical co-existence of both physical and logical assets (including virtual models) that form a cyber-physical system through the interconnection of their virtual representations. The architecture emphasises decentralised interactions and ensures semantics through the integration of a knowledge graph.

The description model of a generalised object transfer system centers around a description of the object state. We describe the requirements for the formation process of the an object transfer system and the process itself through a state transition function. The process is driven by continuous evaluation of the object state. We introduce several important sets of states and demonstrate how the approach can be utilized in different object transfer scenarios.

We implemented an object transfer system based on a vision-controlled autonomous mobile robot and stationary transfer units to realise the flexible transfer of a wide range of objects without the need for a load handling device. The physical assets are also virtually modeled in a physics-engine-based simulation environment.

In our application study, we conduct an extensive set of both real-world and virtual experiments for transferring a 37 unique starter motor cores. The virtual models were initially calibrated using a small set of real-world experiments, resulting in a $F_{0.5}$ score of 0.84 when comparing the binary predictions between the virtual and physical systems. Subsequently, the calibrated models are used as digital twins to predict the most promising pre-transfer state for each individual starter motor core. We validated the results through additional real-world experiments. Despite some inaccuracies, the results successfully demonstrate the potential of digital twins for predicting probabilistic object transfers in a remanufacturing use-case.

Kurzfassung

Um die Auswirkungen der globalen Erwärmung zu mildern, haben Regierungen weltweit Ziele zur Reduktion der Treibhausgasemissionen festgelegt. Um diese Ziele zu erreichen, ist eine sozial-ökologische Transformation des Industriesektors von einem "take, make, waste"-Ansatz hin zu einer Kreislaufwirtschaft erforderlich. Das Remanufacturing, eine der so genannten End-of-Life-Strategien, wird häufig als die effektivste Möglichkeit zur Material- und Energieeinsparung angesehen (Nasr and Thurston 2006). Wirtschaftliche Hürden auf Lieferketten- und Fabrikebene verhindern jedoch eine breitere Anwendung in verschiedenen Industrien. Eine der größten Herausforderungen beim Remanufacturing ist der Umgang mit der hohen Unsicherheit über den Zustand der zurückgegebenen Produkte. Remanufacturing-Prozesse erfordern eine hohe Prozessflexibilität, die mit den derzeitigen Automatisierungsansätzen nicht erreicht werden kann. Industrie 4.0 Methoden werden häufig als Lösungsansätze zur Reduzierung der Unsicherheiten angeführt. Aus Sicht der Forschung besteht jedoch eine Lücke in der erfolgreichen Demonstration von Implementierungen, die die Vorteile dieser Methoden aufzeigen.

Diese Arbeit trägt dazu bei, diese Forschungslücke zu schließen, indem sie eine reale Anwendung von digitalen Zwillingen im Kontext eines flexiblen Übergabesystems für eine Remanufacturing-Anwendung demonstriert. Der Forschungsbeitrag gliedert sich in vier Teile. Im ersten Schritt wird eine nicht-hierarchische Architektur für digitale Zwillinge vorgestellt. Darauf folgt die Einführung eines generalisierten Beschreibungsmodells eines Übergabesystems. Im dritten Schritt wird die reale Implementierung eines Übergabesystems als cyber-physisches System vorgestellt, das sowohl die Architektur als auch das formale

Beschreibungsmodell in einer Demonstrationsumgebung umsetzt. In einer umfangreichen Anwendungsstudie werden die digitalen Zwillinge hinsichtlich ihrer Vorhersagefähigkeit für die Übergabe gebrauchter Startermotoren evaluiert.

Die vorgestellte Architektur zeichnet sich durch eine nicht-hierarchische Koexistenz von physischen und logischen Assets (einschließlich virtueller Modelle) aus, die durch die Verknüpfung ihrer virtuellen Repräsentationen ein cyber-physisches System bilden. Die Architektur ermöglicht dezentrale Interaktionen zwischen allen Assets und stellt eine gemeinsame Semantik durch die Integration eines Wissensgraphen sicher.

Das verallgemeinerte Beschreibungsmodell eines Übergabesystems basiert im Wesentlichen auf der Beschreibung des Objektzustandes. Zunächst wird der Formierungsprozess des verallgemeinerten Übergabesystems dargestellt. Der Übergabeprozess selbst wird als Funktion von Zustandsübergängen des Objekts abgebildet. Während des Übergabeprozesses wird der Objektzustand kontinuierlich validiert. Dazu werden mehrere kritische Zustandsmengen eingeführt, die zur Steuerung des Übergabesystems verwendet werden. Anhand von Beispielen wird gezeigt, dass das Beschreibungsmodell auf verschiedene Übergabeszenarien angewendet werden kann.

In einer realen Demonstrationsumgebung für die Erforschung von neuen Methoden im Remanufacturing wird ein flexibles Übergabesystem implementiert, das aus einem autonomen mobilen Roboter und mehreren stationären Transfereinheiten besteht. Das System ermöglicht die Übergabe einer Vielzahl von Objekten, ohne dass ein Behälter oder ein anderes Lastaufnahmemittel benötigt wird. Die umgesetzten physischen Assets werden zusätzlich in einer physikalischen Simulationsumgebung modelliert und im cyber-physischen System mit den realen Assets verknüpft.

In einer umfangreichen Anwendungsstudie werden sowohl reale als auch virtuelle Experimente für den Transfer von insgesamt 37 verschiedenen gebrauchten Startermotoren durchgeführt. Zunächst werden die virtuellen Modelle anhand

einer kleinen Anzahl realer Experimente kalibriert. Dabei ergibt sich ein $F_{0.5}$-Score von 0,84 für die virtuelle Vorhersage über den Erfolg einer Objektübergabe. Anschließend werden die kalibrierten Modelle als digitale Zwillinge verwendet, um den vielversprechendsten initialen Zustands jedes Startermotors vor der Übergabe vorherzusagen. Die Ergebnisse dieser Vorhersagen werden durch zusätzliche reale Experimente validiert. Trotz einiger Ungenauigkeiten zeigen die Ergebnisse erfolgreich das Potential für den Einsatz digitaler Zwillinge zur Vorhersage unsicherer Objektübergaben in einem Remanufacturing-Szenario.

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Fördertechnik und Logistiksysteme (IFL) des Karlsruher Instituts für Technologie (KIT). Mit der Veröffentlichung dieser Dissertation geht ein sehr lehr- und ereignisreicher Lebensabschnitt zu Ende. Vielen Menschen gebührt ein herzliches Dankeschön, und ich hoffe, dass ich zum Zeitpunkt der Veröffentlichung einem Großteil dieser Menschen bereits einen persönlichen Dank übermittelt habe.

Ein ausdrücklicher Dank gilt meinem Doktorvater Kai Furmans, der mich in den vergangenen fast sechs Jahren vertrauensvoll begleitet hat und mich durch positive Kritik zur internen, fristgerechten Fertigstellung dieser Arbeit motivieren konnte. Auch als langjähriges Mitglied der Institutsleitung fasziniert mich seine Art, durch Eigenverantwortung individuelle Freiräume zu ermöglichen, in deren Grenzen sich jeder frei entfalten kann. Sehr gefreut habe ich mich auch über die Übernahme des Korreferats durch Tullio Tolio, der mir zudem die Möglichkeit gab, seine Forschungsgruppe am Polimi während eines längeren Aufenthaltes kennenzulernen.

Der vielleicht größte Dank gilt den vielen tollen Kolleginnen und Kollegen am IFL, insbesondere meiner ehemaligen ROBIS-Gruppe, die mich 2019 als "Externen" liebevoll in ihren Reihen aufgenommen hat. Euch allen ein herzliches Dankeschön!

Ein weiterer Dank geht an meine vielen Freunde, seien es alte Bekannte aus der Heimat oder inspirierende Musikerkollegen aus der Akademischen Philharmonie Heidelberg, die mir immer wieder Mut zugesprochen und für die nötige

Ablenkung gesorgt haben, die man bei einem Langzeitprojekt wie diesem unbedingt braucht.

Zu guter Letzt möchte ich mich zum einen bei Max bedanken, ohne den mein persönliches Projekt Karlsruhe wahrscheinlich nie zustande gekommen wäre, und zum anderen bei meiner Familie, auf die ich mich immer verlassen konnte und die mich kontinuierlich bei allen Entscheidungen stets unterstützt hat.

Karlsruhe, Januar 2025 Jan-Felix Klein

Contents

Abstract . . . . . **i**

Kurzfassung . . . . . **iii**

1 Introduction **1**
1.1 Problem Description 3
1.2 Scope of the Thesis 5

2 Review on Digital Twins **7**
2.1 Definition of Digital Twins 7
2.2 Functions of DTs 9
2.3 DTs in Production Logistics 9
2.4 Digital Twins in Remanufacturing and Disassembly Automation 11
2.5 Chapter conclusion 17

3 Digital Twin Architecture **19**
3.1 Physical Assets $\mathcal{A}^P$ and Virtual Representations 21
3.2 Logical Assets $\mathcal{A}^{\mathcal{L}}$ and Virtual Models $\mathcal{A}^{\mathcal{V}}$ 25
3.3 Descriptive Layer 26
3.4 Chapter conclusion 27

4 Object Transfer Systems **29**
4.1 Formal Description 30
4.2 Examples 33
4.3 Chapter Conclusion 35

5 Implementation of an Object Transfer System in a Remanufacturing Demonstration Environment . . . 37
5.1 Physical Assets $\mathcal{A}^P$ . . . 39
5.1.1 Transfer Units $\mathcal{TU}^P$ . . . 40
5.1.2 Transport Modules $\mathcal{TM}^P$ . . . 41
5.1.3 Objects $\mathcal{O}^P$ . . . 42
5.2 Logical Assets $\mathcal{A}^L$ . . . 44
5.2.1 Virtual Models $\mathcal{A}^V$ . . . 45
5.3 The Cyber-Physical System . . . 47
5.4 Object Transfer System . . . 50
5.4.1 Global Navigation and Docking Process . . . 51
5.4.2 Object States Definition . . . 53
5.4.3 Detector and Tracker . . . 57
5.4.4 Visual Servoing . . . 60
5.5 Challenges . . . 61
5.6 Chapter conclusion . . . 64

6 Digital Twin Application Study . . . 67
6.1 System Adaptations . . . 69
6.1.1 Pre-Transfer State Restrictions . . . 69
6.1.2 Docking Pose Error Measurement . . . 71
6.2 Experiment Descriptions . . . 73
6.2.1 Real-world Experiment Description . . . 74
6.2.2 Virtual Experiment Description . . . 75
6.3 Calibration Experiments . . . 77
6.3.1 Result Plot Introduction . . . 77
6.3.2 Quantitative Results and Discussion . . . 78
6.4 Synthetic Docking Error Generator . . . 83
6.5 Virtual Model Tuning and Evaluation . . . 85
6.5.1 Simulation Parameter Tuning . . . 87
6.5.2 Instance Parameter Tuning . . . 87
6.5.3 Sim-to-Real Comparison . . . 91
6.6 Virtual Full-State Experiments . . . 93

6.6.1 Determining Best Pre-Transfer State 96
6.7 Validation 100

7 Conclusion 107
7.1 Summary 107
7.2 Outlook 110

A Appendix 113
A.1 Calibration Experiments 113
A.2 Evaluation Metrics 119
A.3 Parameter Sets 119
A.4 Result Tables 122
A.5 Algorithms 126
A.6 Full state experiments result plot 128

List of Acronyms and Symbols 133

List of Figures 141

List of Tables 145

List of Publications 147

Bibliography 151

1 Introduction

Our world is warming, with far-reaching consequences for the environment, ecosystems, economies, and human societies. (UNEP 2020). In the renewed Federal Climate Change Act of 2021, Germany set its greenhouse gas reduction target to at least 65% by 2030 compared to 1990, with the long-term goal of achieving greenhouse gas neutrality by 2045 (Frenz 2022).

To achieve this goal, the industrial sector needs a socio-ecological transformation. The common "take, make, waste" approach of the linear production, see grey path in Figure 1.1, must be transformed into a holistic circular economy that decouples resource consumption from economic growth (Kadner et al. 2021). Figure 1.1 shows different recovery strategies for used products, some of which can be classified as value retention strategies. The common objective of these strategies is either to extend the use phase beyond the intended end of use (EoU) through repair, upgrade or reuse, or to create additional service life beyond the expected end of life (EoL) through remanufacturing or refurbishment. Remanufacturing is often considered to be the most superior strategy in terms of overall material and energy savings (Nasr and Thurston 2006). Today, remanufacturing is a standardised value retention process in which a remanufactured product is created from restored components of one or more used parts as well as new components. Remanufactured products provide at least the functionality and performance of the original product (DIN SPEC 91472:2023-06 2023). The environmental benefits of remanufacturing are embedded in the delay in disposal, the reduction in resource consumption and the savings in embodied energy to manufacture new products (Sundin and Lee 2012).

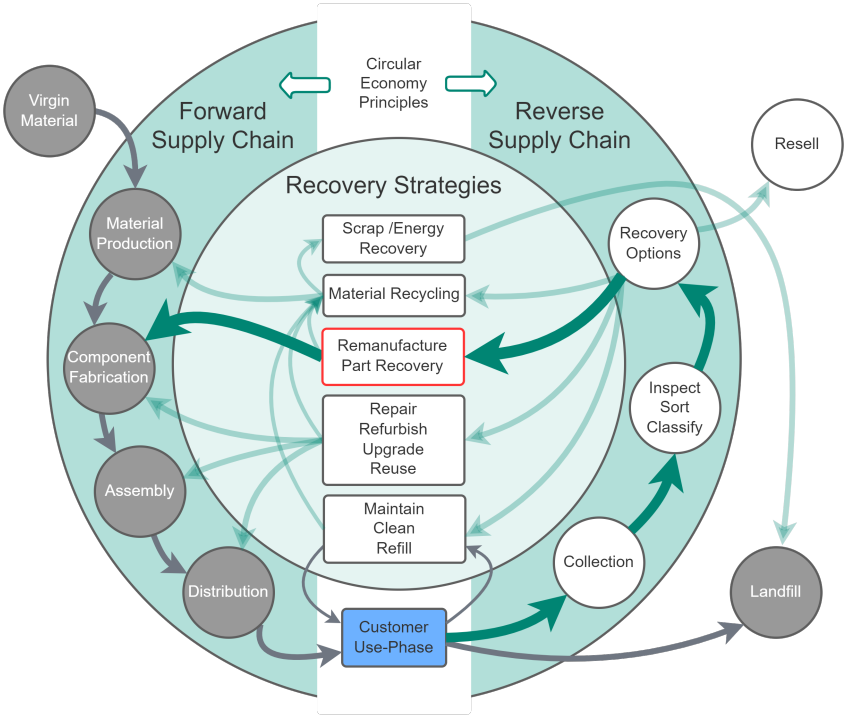


Figure 1.1: Supply chain of a *Circular Product Service System* including different recovery strategies, adapted from (Widmer et al. 2018).

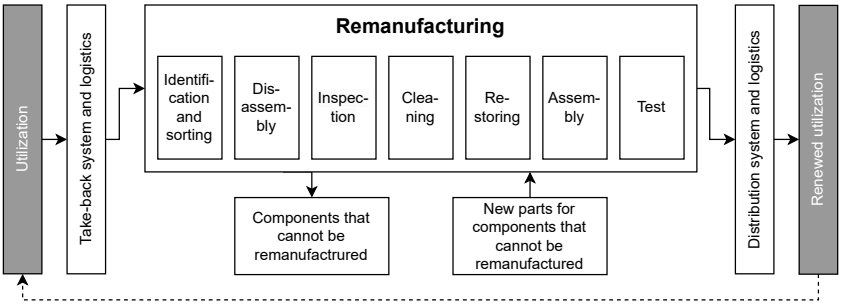


Figure 1.2: The remanufacturing process chain (DIN SPEC 91472:2023-06 2023), similar to (Steinhilper 1998, Matsumoto et al. 2016, Lee et al. 2017).

1.1 Problem Description

Figure 1.2 illustrates the remanufacturing process chain. The primary input to the remanufacturing process are used or non-functional products, assemblies or components. In the manufacturing domain, these used products are commonly referred to as **cores**. The remanufacturing process chain is characterised by a high degree of uncertainty compared to traditional linear production schemes. The core acquisition and collection process is challenged by uncertainty in the timing and volume of returns, as well as uncertainty in the quality of the returned products due to their individual use phases (Wei et al. 2015, Gu and Tagaras 2014). The inherent uncertainty of the product leads to individual process sequences with highly variable lead times, making it difficult to balance returns and demand (Guide 2000). Coping with the variability in quality and quantity requires a high degree of process flexibility at all levels, which until now could only be achieved by a high degree of manual labour with expert knowledge. According to the study by Parker et al. 2015, in 2015, the global ratio of remanufacturing to new manufacturing was only about 1.9% across various industries. Customer recognition, the availability of cores, their quality and high labour costs are mentioned as the main barriers. Researchers generally suggest that increased automation and the implementation of Industry 4.0 methods to reduce costs and improve productivity and quality are potential drivers for the wider application of remanufacturing (Kurilova-Palisaitiene et al. 2018, S. Yang et al. 2018, Kerin and Pham 2019, Teixeira et al. 2022). However, traditional automation methods, which rely on a high degree of determinism, are not feasible due to the uncertainty inherent in the product and processes. As a result, most of the research in this context has been conceptual. The lack of real-world demonstrations that show the benefits of applying advanced methods that can cope with the uncertainty in remanufacturing is subsequently identified as a research gap (Kerin and Pham 2019, Tiwari et al. 2021, Teixeira et al. 2022). This thesis aims to contribute in filling this research gap. As part of the overall contribution, we demonstrate how to implement and successfully use digital twins, one of the key Industry 4.0 technologies, to reduce process uncertainty

for a real-world use case. In the following, we briefly introduce the contributions and research questions addressed in this thesis:

As an initial contribution, we propose a generic digital twin architecture embedded in a cyber-physical system. The architecture is characterised by a non-hierarchical network of smart, encapsulated assets and their corresponding virtual models, as well as a semantic description layer to ensure interoperability. The corresponding research question is:

> **How can we design a digital twin architecture that emphasises decentralised machine-to-machine interactions?**

In our application study, we investigate the process of transferring cores without any load carriers. Since the variant of the returned core is uncertain, this object transfer process is a suitable example to demonstrate the prediction capabilities of digital twins in a remanufacturing use case. Since, to the best of our knowledge, there is no common description of an object transfer system, we formulate an abstract, system-independent one. The corresponding research question is:

> **How can we formally describe an object transfer system?**

We combine both the digital twin architecture and the object transfer description in the real-world implementation of a cyber-physical system in which the implemented assets physically realise the object transfers in a remanufacturing demonstration environment. The related research question is the following:

> **How can we implement a cyber-physical system that allows the automated object transfer in a remanufacturing environment?**

We use the implemented cyber-physical system in an extensive application study to explore the capabilities of the digital twins in predicting the probability of success for the object transfer of a set of starter motor cores. The corresponding research question is:

> **How can we utilize digital twins to predict the probability of success for the object transfer of used starter motor cores?**

1.2 Scope of the Thesis

This thesis is divided into seven chapters. Following the motivation and problem description in Chapter 1, Chapter 2 provides an overview on digital twins and their applications in production logistics and material handling. In a structured literature review, we analyse the state of the art of digital twins used in remanufacturing and disassembly automation. Chapter 3 presents a non-hierarchical digital twin architecture that emphasises decentralised machine-to-machine interactions. Chapter 4 introduces a formal description of an object transfer system and provides illustrative examples. Chapter 5 presents the implementation of a cyber-physical system to realise both real-world and virtual object transfers as part of a remanufacturing demonstration environment. Chapter 6 covers an extensive application study in which we calibrate the virtual models and validate their predictive capabilities. Finally, Chapter 7 summarises the contributions and provides an outlook for future research.

2 Review on Digital Twins

In this chapter, we first review different definitions of digital twins in Section 2.1 and summarise functions of digital twins in Section 2.2. We examine examples in the production logistics domain in Section 2.3 and conduct a structured literature review on digital twins in remanufacturing and automated disassembly in Section 2.4. Finally, we summarise the review results in a Section 2.5.

2.1 Definition of Digital Twins

The general concept of a *Digital Twin* (DT) was introduced in the work of Grieves (2005) in the context of *Product Lifecycle Management*. In his work, Grieves describes the *Mirrored Space Model*, which consists of three elements, the real space, the virtual space(s), and a bidirectional linking mechanism in between. One of the first mentions of the term *Digital Twin* appeared in a NASA roadmap where it was used as a synonym for an integrated multiphysics, multiscale, probabilistic simulation of an as-built system with the goal of mirroring the real system in the best possible way (Shafto et al. 2012, Glaessgen and Stargel 2012). Over the years, the increased interest in DTs in different domains has led to many different definitions as well as an increasing number of reviews that try to give a clear overview of the existing ones. For example, VanDerHorn and Mahadevan (2021) compared 46 definitions found in the literature over a period of 10 years and proposed a consolidated definition that describes a DT as *"a virtual representation of a physical system (and its associated environment and processes) that is updated through the exchange of information between the physical and virtual system"* (VanDerHorn and Mahadevan 2021). The

growth in different definitions is driven by the usage of the term *Digital Twin* as a marketing buzzword for any technology that relies on simulations. To date, most definitions are unified by the same three concepts that make up a digital twin, a physical system/entity/asset, a virtual representation/model and a bidirectional information transfer between them. Kritzinger et al. 2018 emphasise this bidirectional information transfer as the main characteristic that distinguishes a DT from the closely related concepts of a digital model and a digital shadow. In the manufacturing domain, another commonly referred definition is given by Tao et al. 2018, who extends the triple concept by two additional dimensions, a data model and a service mode. Standardisation efforts towards a common understanding of digital twins are still at an early stage (Ferko et al. 2023).

In this thesis, we use the terms *physical twin* and *physical asset* interchangeably to refer to the real-world system and *virtual model* to refer to its virtual twin. We agree with Tao et al. 2018 that the simple triple concept does not satisfy the expected functionality of a digital twin and refer to a DT as a:

> virtual model of a physical twin, coupled with an analytics engine that influences the decision-making process of the physical twin through bidirectional communication.

This DT concept is visualised in Figure 2.1.

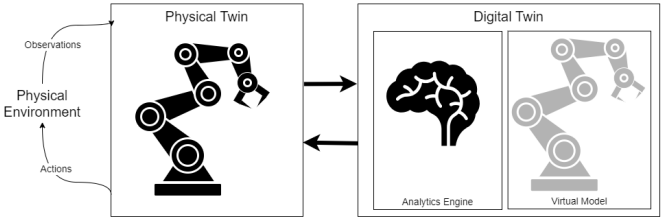


Figure 2.1: *Digital Twin* concepts: A physical twin observes and acts in its physical environment. The digital twin consists of a virtual model that mimics its physical twin and an analytics engine that provides decision-making capabilities. The digital twin is directly connected to its physical twin in a bidirectional manner.

2.2 Functions of DTs

A DT is primarily used as a decision support tool. As illustrated in Figure 2.1, a digital twin consists of a virtual model and an analysis engine that controls the interface between the physical and the digital twin based on the desired functionality. We distinguish three main functions of a DT, similar to Glatt et al. 2021:

- **Diagnosis**: During diagnosis, the DT reconstructs past states of the physical twin to better understand and re-evaluate past decisions or to re-tune its virtual models to improve their quality.
- **Monitoring**: During monitoring, the virtual model is used as a shadow of the physical twin and mirrors the state of the physical twin. Besides being used as a visualisation tool for human supervision, intelligent algorithms can use the mirrored state to detect anomalies and influence the behaviour of the physical twin.
- **Forecasting**: Forecasting generally involves predicting future states of the physical twin. These predictions can be based purely on past states of the physical twin, or by using the virtual model to calculate future states based on expected future input values. Examples include predicting the probability of a mission success, or evaluating system responses to possible scenarios before they become critical to the physical twin.

2.3 DTs in Production Logistics

In this section we review some selected publications on DTs in production logistics that relate to the later chapters of this thesis. Kaiblinger and Woschank (2022) and Zhu et al. (2023) both provide recent comprehensive literature reviews on the application of DTs in production logistics and we refer to them

for a broader overview. In general, Zhu et al. (2023) distinguishes five different application scenarios of DTs in the production logistics process, namely transportation, packaging, warehousing, material distribution and information processing. Hauge et al. (2020) shows how digital twins can help in the decision-making process of selecting and comparing the appropriate equipment for a given task. In one example, the authors compare the paths generated by different types of *Automated Guided Vehicles* (AGVs) to decide on which vehicle is best suited to the layout. Stączek et al. (2021) lists several benefits of using digital twins for *Autonomous Mobile Robots* (AMRs) and AGVs, e.g., for developing navigation and localisation algorithms, testing safety functions, predicting reliability, etc. In a case study, the authors show how an AMR system can be virtually tested for a new scenario at an early stage of a new integration project. Vachálek et al. (2021) builds a virtual model of a conveyor belt feeding three workstations with the aim of using the digital twin as a test platform for newly developed control algorithms. The authors verified the implementation in a simple real-world test environment. Kaiblinger and Woschank (2022) conducted a full-text analysis of 24 papers on digital twins in the context of production logistics. Among other things, the authors compared the technologies used to implement the virtual models of the DTs. While the majority relied on *Discrete Event Simulation* (DES), only three papers made use of a physical simulation in which the virtual models are embedded. C. Hu et al. (2020) conceptually explores different application areas of DTs in the manufacturing of complex electronic equipment while considering the entire product life cycle. However, the use of physical simulation is only mentioned during the operation and maintenance phase of the products under consideration. Zheng et al. (2019) virtually models a welding production line using physical and kinematic models. The study mainly focuses on the data transfer and mapping from the physical twin to to the virtual model. Tao et al. (2018) is mentioned as the third study using physical models, but as it's not directly related to production logistics, it will not be further discussed here.

A closely related contribution was proposed by Glatt et al. (2021). The authors implemented a DT of a material handling system that moves workpieces placed

on a workpiece pallet along a single axis. The virtual model is used to estimate appropriate control parameters to ensure a disturbance-free motion. In a preliminary study, the authors compared the fidelity of the Bullet physics engine with an analytical model of a workpiece tipping due to the braking deceleration of an AGV (Glatt et al. 2019).

2.4 Digital Twins in Remanufacturing and Disassembly Automation

The content of this section extends upon a working paper titled "A Study on the Predictive Capabilities of Digital Twins for Object Transfers in Fluid Automated Remanufacturing" (Klein and Furmans 2025). The author of this thesis was responsible for the conceptualisation, methodology, software programming, validation, formal analysis, writing, and visualisation of the research presented in this section.

This section presents a structured literature review at the intersection of DTs, remanufacturing and disassembly automation. The review covers the state-of-the art on existing approaches of DT in the remanufacturing context. The process of selecting the reviewed literature is illustrated in a PRISMA chart in Figure 2.2. We conducted a full-text review of 21 of the initial 42 identified records given the query displayed in the figure caption. We structured the reviewed records according to the following four characteristics:

- **Contribution Type**: Contributions in the field of DTs can take various forms. We evaluate the contribution type by assigning one or more of the following aspects: *"Review"*, *"Model"*, *"Product Model"* and *"Algorithm"*.
- **Assets**: The reviewed publications address various physical assets. During the review, we found three main types of assets, namely *"Core"*, *"Waste"* and *"Resources"*. A category assigned to a record in brackets refers to only a minor contribution to that assets type.

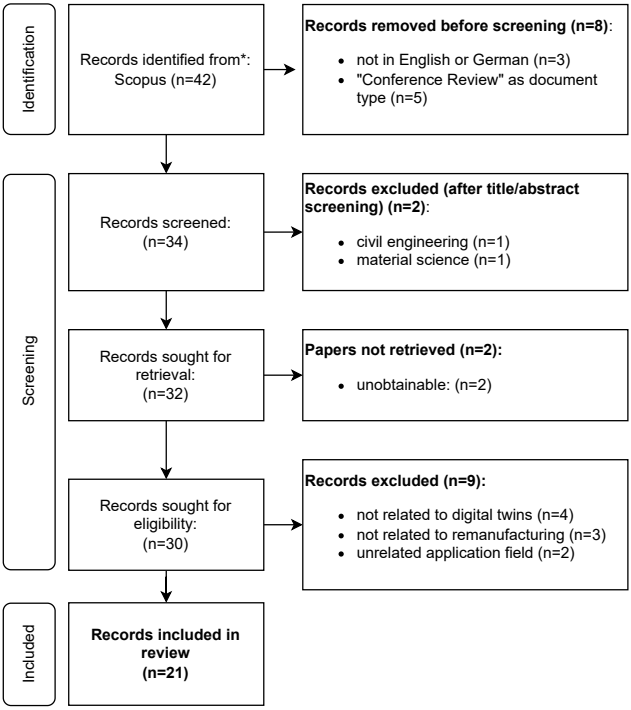


Figure 2.2: PRISMA chart of the structured literature review. The records were retrieved from Scopus on the 26th of March 2024 using the query *(digital AND twin*) AND ((remanufactur*) OR (disassembly AND automation))*.

- **Scope**: DTs can be implemented with varying levels of granularity. We differentiate the scope of the DTs into the following categories: *"Life Cycle"*, *"Supply Chain"*, *"Reconditioning"*, *"Disassembly"*, and *"Shop Floor"*.

- **Real-World Demonstration**: The lack of real-world demonstrations of Industry 4.0 methods in remanufacturing is frequently identified as a research gap (Kerin and Pham 2019, Tiwari et al. 2021, Teixeira et al. 2022). Therefore, we rate each record based on its real-world demonstration using

a scale from "-" to "+++". A "-" indicates no real-world demonstration, while "+++" signifies an extensive application study.

Table 2.1 shows the classification results based on our analysis. In the following, we briefly discuss each publication. We structured the discussion of the individual records according to the different scopes.

Table 2.1: Categorised research publications on digital twins in the context of remanufacturing and disassembly automation.

Reference	Contribution Type	Assets	Scope	Real-World Demonstration
Wang and Wang 2019	Architecture, Product Model	Core	Life Cycle	+
Rocca et al. 2020	Type	Core, Resources	Disassembly	++
Tozanli et al. 2020	Model	Core	Life Cycle	-
Wang et al. 2020	Architecture	Core, (Resources)	Life Cycle	-
Chen and Huang 2021	Review	Core, Waste	Supply Chain	-
Shrivastava et al. 2021	Analysis	Core, Resources	Reconditioning	-
Y. Yang et al. 2021	Model	Waste	Shop Floor	+
Zacharaki et al. 2021	Architecture	Core, Resources	Life Cycle	-
Assuad et al. 2022	Architecture	Core	Shop Floor, Reconditioning	+
Ghorbani and Khameneifar 2022	Model, Algorithm	Core	Reconditioning	+
Guo and Han 2022	Model	Core	Supply Chain	-
Y. Hu et al. 2022	Architecture	Core, Resources	Shop Floor	-
Lanza et al. 2022	Architecture	Core, Resources	Shop Floor	-
Wiesner et al. 2022	Model	Core	Life Cycle	+
Elsner et al. 2023	Architecture	Core, (Resources)	Life Cycle	-
Ke et al. 2023	Architecture, Model	Core	Life Cycle	+
Kerin, Hartono et al. 2023	Model	Core	Life Cycle, Shop Floor	-
Kerin, Pham et al. 2023	Product Model	Core	Life Cycle	-
Klein and Furmans 2023	Architecture, Model	(Core), Resources	Shop Floor	+
Sun et al. 2023	Architecture	(Core), Resources	Supply Chain	+
Li et al. 2024	Architecture, Model	(Core), Resources	Disassembly	++
This thesis	Architecture, Model, Algorithm	(Core), Resources	Shop Floor	+++

Life Cycle

The majority of the literature reviewed focuses on DTs from a life cycle perspective. Wang and Wang 2019 propose a system architecture for a digital twin-enabled *Cyber-Physical System* (CPS) and a data model that considers data from the design phase through manufacturing, use phase and recovery over multiple life cycles. An exemplary implementation shows a cloud-based system where core data can be accessed via a web-based front-end by scanning an attached QR code. Tozanli et al. 2020 propose a digital twin-based model to estimate an accurate take-back price for a core based on historical data. A discrete event simulation is used to compare different trade-in policies. Wang et al. 2020 provide a holistic architecture that covers multiple product life cycles and attempts to include all stakeholders involved in the remanufacturing process, with the aim of demonstrating a big data application. However, the architecture is highly conceptual and the application scenario as well as the assumption of the potential benefits of applying the architecture are not based on real-world data. Similarly, Zacharaki et al. 2021 present the RECLAIM (*REmanufaCturing and Refurbishment LArge Industrial equipment*) architecture of the EU Factory of the Future project. While the holistic architecture is not discussed in greater detail, the publication emphasises the decision support framework based on a digital twin of the machinery with the aim of optimising maintenance activities. Wiesner et al. 2022 present a data-driven approach to assessing the health of a machine component. For their use case, the authors developed a custom sensor box to predict the remaining time to failure of a spindle bearing. Another architecture with life cycle perspective is proposed by Elsner et al. 2023. The goal of the architecture is to enable dynamic life cycle assessment in the remanufacturing of battery modules. Ke et al. 2023 present an intelligent redesign methodology for used products and applies it to clutch case study to find the optimal clutch redesign scheme. Kerin, Hartono et al. 2023 propose a decision making model that optimises the remanufacturing planning while using data from different stages in a product's life cycle. The authors use a neural network to predict the remaining useful life and a bees algorithm to plan the disassembly sequence. In another publication Kerin, Pham et al. 2023 propose a generic

product model. After defining the requirements for a product DT, the authors propose a UML class diagram and instantiate it for a single example product.

Supply Chain

Three publications examine DTs in the context of remanufacturing from a supply chain perspective. Chen and Huang 2021 review the problem of information sharing and examines information asymmetries in remanufacturing supply chains. Guo and Han 2022 propose a closed-loop digital twin supply chain framework and evaluates the impact of digital twins on the bullwhip effect. Sun et al. 2023 propose an architecture for a digital reverse logistics twin in the context of remanufacturing network design. The architecture is evaluated in a case study of a remanufacturing design problem in Norway.

Reconditioning

A further three reviewed publications focus on reconditioning. Shrivastava et al. 2021 address the challenges of additive manufacturing in remanufacturing. The authors mention digital twins in combination with other data-driven models as a possible solution to limit the initial process parameter space and reduce of trial-and-error testing. Assuad et al. 2022 propose a decision framework for a remanufacturing system based on a laboratory use case, while emphasising the reconditioning process chain. However, the utility and the interface to specific virtual models of a DT is not further explained. Ghorbani and Khameneifar 2022 propose a methodology to construct accurate, damage-free digital twins of defective turbine blades and an algorithm to register noisy scanned data to achieve accurate repair volumes for the additive manufacturing process.

Disassembly

Rocca et al. 2020 provide a comprehensive study of Industry 4.0 technologies in the context of circular economy. The authors developed two simulation models, one to optimise the design of the manufacturing system and a second to optimise the energy consumption during the disassembly process. They also practically demonstrated the capabilities of the DT in a laboratory experiment. Li et al. 2024 implemented a human-robot collaboration system and demonstrated the prototype on an electric vehicle battery disassembly task. The system is based on a visual reasoning module that infers task planning strategies while respecting safety and ergonomic rules.

Shop Floor

Y. Yang et al. 2021 propose a model to dynamically forecast the amount and timing of waste generated during the disassembly process. The model incorporates a digital twin learning loop, in which the model is continuously updated based on new information. The architecture of Y. Hu et al. 2022 is the closest to the one presented in the following chapter. The authors propose a three-layer architecture for a cyber-physical remanufacturing system, consisting of a physical layer, an edge layer and a cloud service layer. In contrast to the architecture proposed in Chapter 3, the framework provides an hierarchical approach in which the digital twins are on the edge layer and the logical assets that control the processes are on the top cloud service layer. The proposed architecture by Y. Hu et al. 2022 is conceptual only, with no real-world implementation or application provided. Both Lanza et al. 2022 and Klein and Furmans 2023 are directly related to this dissertation and will be discussed and referenced in the later chapters.

2.5 Chapter conclusion

The content of this section extends upon a working paper titled “A Study on the Predictive Capabilities of Digital Twins for Object Transfers in Fluid Automated Remanufacturing” (Klein and Furmans 2025). The author of this thesis was responsible for the conceptualisation, methodology, software programming, validation, formal analysis, writing, and visualisation of the research presented in this section.

As one of the key technologies of the Industry 4.0 era, digital twins have received a lot of research attention over the last decade. The inflationary use of the term makes it difficult to provide a consistent definition and to distinguish it from well-known and much older concepts. We refer to a DT as a virtual model of a physical twin, coupled with an analytics engine that influences the decision-making process of the physical twin through bidirectional communication. In the production logistics, most existing DT applications are demonstrated from a system-level perspective, where a single virtual model is used to abstract the entirety of a production logistics system. The majority of conducted research on DTs in the remanufacturing and disassembly automation domain focuses on the life cycle and supply chain perspective, there are only a few application studies at the shop floor level. As noted in (Chen and Huang 2021) and confirmed by the conducted analysis, there is only little applied research on digital twins in remanufacturing that has been validated in a real-world application. As indicated in the last row of Table 2.1, this thesis contributes in closing this research gap by providing a real-world implementation and an extensive application study to validate the capabilities of the DTs.

3 Digital Twin Architecture

There is a wide variety of different DT architectures proposed in the scientific literature. While patterns are emerging, there is still no commonly accepted architecture yet (Karabulut et al. 2024, Ferko et al. 2023). The architecture proposed by Ashtari Talkhestani et al. 2019 consists of a unique ID, models and associated interfaces to tools, version management, operation data of the physical twin, organisational and technical data, information about the relations to other DTs and a communication interface. The authors further define an extension of the aforementioned components, which they refer to as an *intelligent* DT. The additional modules include model comprehension, algorithms to implement intelligent functionality, a service and a feedback interface. In the architecture proposed by Souza et al. 2019, the authors emphasise an IIoT gateway between the physical and digital twins, as well as different views for different end users. (Minerva et al. 2020) review different IoT architectures and how they can be beneficially used to enable digital twins. Redelinghuys et al. 2020 introduce a six-layer architecture that ranges from the physical system or physical twin to the cloud while emphasising the non-hierarchical connection of the emulation and simulation layers. Steindl et al. 2020 present a simple, technology-independent architecture based on the 5D architcture by Tao et al. 2018, that is aligned with the RAMI 4.0 metamodel. The architecture includes a shared knowledge base as a central point of information and the *Asset Administration Shell* (AAS) as a promising way to standardise data representation.

Unlike the architectures described above, the architecture presented in this chapter has a different focus. The architecture is based on the vision of a fully autonomous, intelligent and self-reconfiguring factory. In this vision, individual, autonomous assets form a loosely interconnected CPS. The physical

assets have the ability to autonomously sense, analyse their surroundings, and respond to changes in their environment. The assets synergise their capabilities to collaborate through decentralised interactions, much like human workers on the shop floor who have their own goals and tasks, but who coordinate and support each other when facing uncertainties. The CPS as a whole aims to optimise value creation processes. The reconfigurability of the CPS enables it to implement constantly changing master production schedules. The following characteristics were driving the architecture design:

- **Asset-centered**: Physical and logical assets are in the core of the architecture, they coexist without a hierarchy.
- **Decentralised**: Assets are able to establish communication channels among each other without a central entity, enabling decentralised decision making.
- **Enriched by semantics**: A descriptive layer ensures semantic consistency and allows assets to reason about the data models of other virtual representations.

The above characteristics enable a dynamic CPS with plug-and-play capabilities in which assets continuously being able to join and leave the system. The proposed architecture in this chapter is not holistic in the sense that it does not necessarily encompass the entire lifecycle of the assets and does not explicitly consider organisational interfaces. We model the architecture in a step-by-step approach in the following sections which expands our previous work in (Enke et al. 2022) and (Klein and Furmans 2023). Section 3.1 introduces the concepts of physical assets and virtual representations. Section 3.2 continues by introducing logical assets and virtual models. Section 3.3 gives an insight into the descriptive layer and its connection to the virtual representations of the assets. Section 3.4 concludes the chapter.

3.1 Physical Assets $\mathcal{A}^P$ and Virtual Representations

Combining the definitions in (Plattform Industrie 4.0 2022) and (Qi et al. 2021), we define a physical asset as a real-world object that

- obeys physical laws and rules and is surrounded by uncertainties,
- is owned by or under the custodial duties of an organisation,
- has perceived or actual value to the organisation.

In the manufacturing and logistics domain, physical assets include, e.g., machines or workstations on a shop floor, material handling systems, and infrastructure such as laboratories, offices or open space on a factory floor. The *Virtual Representation* of an asset represents the asset at the cyber layer and provides the interfaces for digital interaction with the asset. Multiple assets providing a virtual representation can form a CPS (Monostori et al. 2016). Communication in a CPS is typically achieved by using either a request-response or publish-subscribe pattern to exchange data following one of several industrial communication protocols (Wollschlaeger et al. 2017). The virtual representation serves two primary functions. It allows other assets to read information from the asset or to actively impact the asset, e.g, by writing a variable, triggering of a method or starting an event. We distinguish between five types of physical assets based on their relationship to their virtual representation, see also Table 3.1:

- **Type A**: These assets have no virtual representation and are not digitally connected. Examples may include e.g. infrastructure components, such as pillars or simple doors on the factory floor
- **Type B**: These assets have a virtual representation but cannot synchronise dynamic state information on their own. Examples may include workpieces, workpiece carriers, or containers.

Table 3.1: Comparison of different asset types.

Type	Virtual Representation	State Synchronization	External Control Interface
A	no		
B	yes	external	no
C	yes	internal	no
D	yes	external	yes
E	yes	internal	yes

- **Type C**: These assets are able to synchronise dynamic state information on their own. Examples include smart shelves that use sensors to detect their contents, or manually operated forklifts that continuously update their current position.
- **Type D**: These assets provide an external control interface and but are unable to synchronise dynamic state information on their own. An example is a stationary conveyor module that provides its speed control externally, but has no information about whether an object is currently present on the module.
- **Type E**: These assets are capable of both self-synchronising dynamic state information and providing an external control interface. An example would be an AMR that updates its pose information and provides interfaces to queue a transport task.

Figure 3.1 visualises the different asset types including their virtual representations, their ability to synchronise their state and the provided control interfaces.

In recent years, there have been increased standardisation efforts to define a common data structure for virtual representations. A common approach for the implementation of a virtual representation is the *Asset Administration Shell*

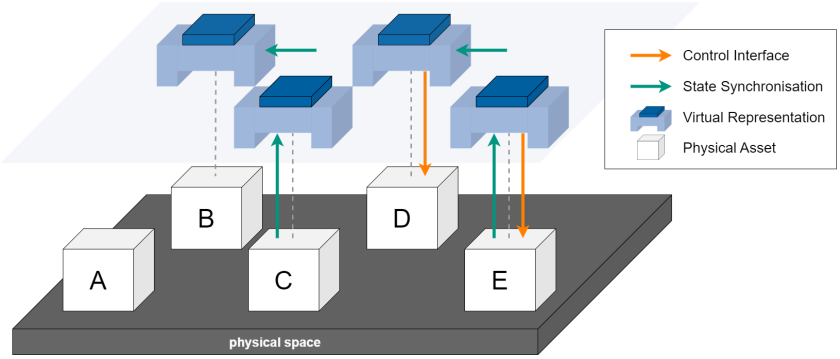


Figure 3.1: Different types of physical assets and their virtual representations.

(AAS) (Plattform Industrie 4.0 2022). The AAS defines a technology independent data model which increases the interoperability between different assets by defining a standardised data structure (Bader and Maleshkova 2019). Each AAS consists of a header containing identification details and a body which comprises one or more submodels that encapsulate specific aspects of an asset. Each submodel is made up of one or more so-called submodel elements. Different types of submodel elements are specified, ranging from a single property data fields to file exchange methods or triggerable events.

Encapsulating a physical entity, such as a mechatronic system, into a physical asset raises the question of how to define system boundaries. From our experience, this decision must be made anew for each individual system and use case. Taking an AMR as an example, on the one hand, we could define the entire AMR as a physical asset, or break it down into individual components, such as the laser scanner, or the drives. While both approaches are valid, a separation seems only useful if other assets exist that need to directly interact with the component.

We split the physical asset into four different layers which are interconnected by an internal communication scheme, see Figure 3.2. The bottom layer consists of the physical components of the asset, which may include actuators, sensors and other devices. The *Hardware Abstraction Layer* (HAL) separates core functionality from hardware details, hides differences in hardware and improves

portability (Jorg et al. 2014). With regards to our AMR example, the HAL abstracts sensors and actuators like motors, encoders, bumpers or lasers (Pont et al. 2005). The *Control Layer* provides basic functionality for applications. In case of an AMR, this may include software modules for path planning, motion control or localisation. The *Application Layer* is the main interaction gateway to the virtual representation of the asset. It provides advanced functionalities, that combine the basic functions provided on the control layer. In the case of an AMR, this layer may provide the functionality to move to a specific target position or to fulfil a transport task. The *Modular Material Handling Framework*, see (Furmans and Gue 2018) defines a set of elementary functions for decentralised material handling systems that, when combined, can map the majority of material handling activities. These functions can also be seen as examples for both, standardisation efforts for virtual representations and executable functions on the application layer. A detailed implementation examples of a physical asset and its virtual representation is presented in Chapter 5.

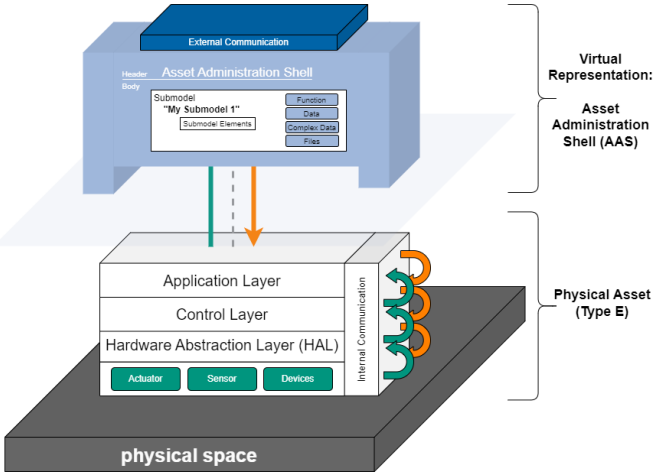


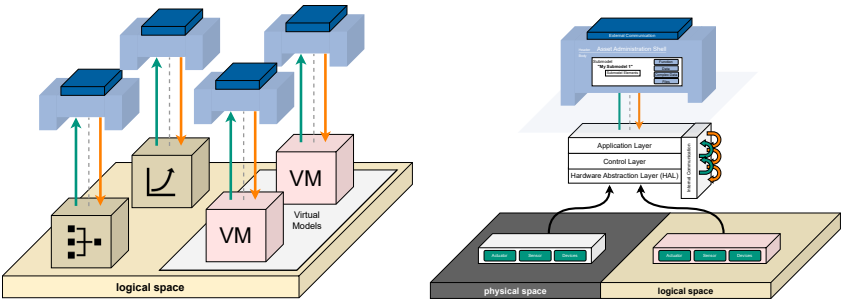
Figure 3.2: Detailed view of a type **E** asset and its virtual representation.

3.2 Logical Assets $\mathcal{A}^{\mathcal{L}}$ and Virtual Models $\mathcal{A}^{\mathcal{V}}$

Unlike physical assets, logical assets are not real-world objects, but software modules that exist as independent entities in the CPS. Referring back to Section 2.2, a minimum set of logical assets is required to implement a DT:

- a virtual model, a digital clone of the physical twin,
- a logical asset that implements the analysis engine, manages the interaction between the physical asset and the virtual model, and ensures the desired function of the DT,
- a logical asset that models the interaction between the physical twin and its environment, this includes simulating time.

Figure 3.3a shows four logical assets. Two of them are virtual models, embedded in the logical space. In general, logical assets can implement many different functions in a CPS. Examples include high-level process controllers, analysis tools, human-machine interfaces or databases. We define virtual models as a special subset of the logical assets ($\mathcal{A}^V \subset \mathcal{A}^L$). Since a virtual model is a digital clone of a physical asset, it is highly recommended that the virtual model is structured as identically as possible to its physical twin. In the best case scenario, the virtual representation and a large number of asset-specific software modules can be used for both the physical asset and its virtual model. Figure 3.3b illustrates this case where only the lowest layer of the asset is different. For the AMR example, this can be achieved by virtually simulating all necessary sensor data as well as the movement of the vehicle over time, given the calculated control signals.



(a) Logical assets and virtual models

(b) Virtual model and physical twin

Figure 3.3: Logical Assets.

3.3 Descriptive Layer

The descriptive layer is the semantic backbone of the architecture. In a distributed environment, the data provided or expected by the virtual representations of two unrelated assets may be heterogeneous in syntax, schema or semantics (Cho et al. 2019). In the presented architecture, the main objective of the descriptive layer is to support machine-to-machine interaction, which is particularly important to enable plug-and-play capabilities where the entities of the CPS are able to automatically reason about the functionality of other assets. The descriptive layer is usually implemented as a *Knowledge Graph* (KG). A KG is a directed graph composed of nodes and edges, where a node represents an entity, that is either a real object or an abstract concept, and an edge that represents the semantic relationship between the two entities (Peng et al. 2023). The schema for a KG is commonly defined as an ontology, which is a concrete, formal representation of what terms mean in the context in which they are used (Hogan et al. 2021). Karabulut et al. 2024 recently conducted a structured literature review to better understand how ontologies are used in the context of DTs. The authors conclude that ontologies are becoming increasingly popular for facilitating KGs in DTs. Future research directions include deeper integration of ontologies in DTs, adoption of ontologies across domains, using a KG as a state graph rather

then simply for storing metadata, and increased use of KGs as part of reasoning processes.

3.4 Chapter conclusion

Figure 3.4 shows the combined DT architecture. The architecture distinguishes between two main types of assets, physical assets and logical assets, with virtual models being a special type of logical assets. All assets form a CPS by exposing their virtual representation on the cyber layer and allow to establish communication channels in between them. Assets can be distinguished by their ability to synchronise their state and exposing a control interface. A knowledge graph on the descriptive layer acts as the semantic backbone of the architecture and is used to establish semantic relationships between the information provided on the virtual representations. Chapter 5 provides a detailed implementation case of the architecture.

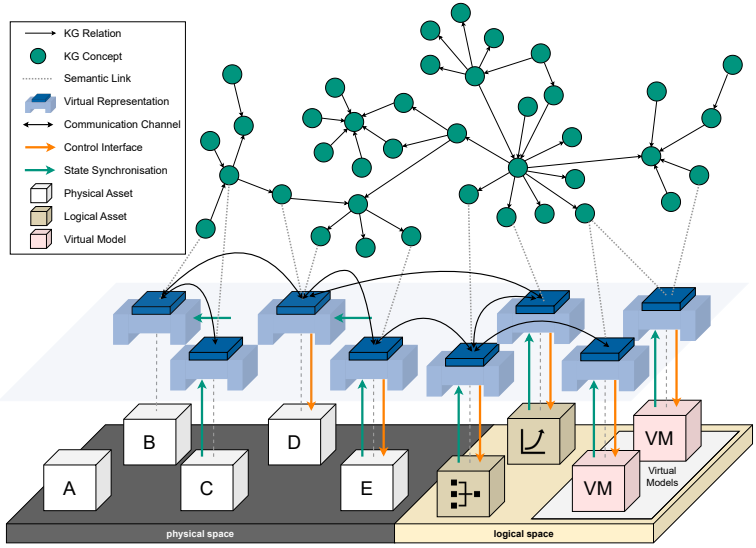


Figure 3.4: The proposed *Digital Twin* architecture.

4 Object Transfer Systems

An object transfer describes a special type of material handling operation. The main characteristic of an object transfer operation is the change of possession of the object under consideration. From a human perspective, an object transfer is a special form of a joint action, in which two or more individuals coordinate their actions in space and time to bring about a change in the environment (Sebanz et al. 2006). In the field of robotics, there is a branch of research that focuses on object handovers from robot to human or from human to robot. This interaction is described as a sophisticated joint action of a giver transferring an object to a receiver, involving many different aspects such as communication, planning, grip release, etc. (Ortenzi et al. 2021). In the field of material handling, there exists only a few contributions that touch the concept of an object transfer. Furmans and Gue 2018 introduce the *Modular Material Handling Framework* which specifies a set of elementary physical functions that, when combined, can cover the majority of material handling operations. The *transfer* function describes the movement of an object from one module to another and differs from the *move* function due to its implication of a changing possession. Elfaham and Epple 2020 define the *handover* concept. Handover abilities include *put*, *pick*, *pass*, *receive*, *release* and *retrieve*. The authors list conditions that must be met to initiate a handover. The conditions include a position overlap, matching of handover abilities by the partners involved, and the matching of physical parameters. Both of these meta-models only superficially describe concepts related to object transfer. In this chapter we introduce a generic description model that can be used as a basis for many different object transfer applications. We use the model in the chapters 5 and 6 for a specific use case.

The chapter is organised as follows. Section 4.1 introduces a formal description for an object transfer system in its simplest form. Section 4.2 shows three different examples in which we apply the description to real-world systems. Section 4.3 concludes the chapter.

4.1 Formal Description

We define an *Object Transfer System* (OTS) as a *System of Systems* (SoS). In its simplest form, the system consists of three assets $(o, a_{\text{pre}}, a_{\text{post}})$:

- $o \in \mathcal{O} \subset \mathcal{A}$ describes a single object to be transferred. We assume the existence of various object instances that can be transferred. We denote the set of all these objects as $\mathcal{O}$.
- an asset $a_{\text{pre}} \in \mathcal{A} \setminus \mathcal{O}$ that initiates the transfer and is in possession of o at the time the OTS is formed,
- an asset $a_{\text{post}} \in \mathcal{A} \setminus \mathcal{O}$ which is supposed to possess o after a successful transfer.

For each individual transfer, the OTS is re-initiated, meaning that the triplet $(o, a_{\text{pre}}, a_{\text{post}})$ is formed, and the system boundaries are established accordingly. However, this does not entail a physical rearrangement of assets. For example, in the case of continuous handling equipment, the system remains stationary, and no physical relocation is necessary for each transfer. Once the transfer is complete, the OTS is dissolved. The initiation process for forming the OTS is typically, but not always, carried out by a_{pre}.

We describe the transfer process itself by means of the state of o. The state definition of o can be multidimensional and can vary greatly depending on the system under consideration. We define the state of o as a vector s consisting of n components denoted as x_i, where $i = 0, 1, 2, \ldots, n$. The first component represents the asset in possession of o ($x_0 = a$), and the remaining components,

$x_1, x_2, \ldots, x_n$, may vary depending on the system. These components may include numerical values such as pose information or categorical data. The set $\mathcal{S}_o$ comprises all possible states of o during the lifetime of the OTS.

$$\mathcal{S}_o = \left\{ s = \begin{pmatrix} a \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} : a \in \{a_{\text{pre}}, a_{\text{post}}\} \right\} \tag{4.1}$$

We divide $\mathcal{S}_o$ into two exclusive subsets based on their ownership property:

$$\mathcal{S}_o^{a_{\text{pre}}} = \{(a_{\text{pre}}, x_1, x_2, ..., x_n)\} \tag{4.2}$$

$$\mathcal{S}_o^{a_{\text{post}}} = \{(a_{\text{post}}, x_1, x_2, ..., x_n)\} \tag{4.3}$$

Before the object transfer process can be started, o must be in a valid starting state. We further call these valid states *Pre-Transfer States*. Formally, we say that there exists a predicate $P_{\text{pre}}(s)$ which evaluates whether a state s satisfies all conditions to be considered a pre-transfer state for the OTS.

$$\mathcal{S}_o^{\text{pre}} = \{s \in \mathcal{S}_o^{a_{\text{pre}}} : P^{\text{pre}}(s)\} \tag{4.4}$$

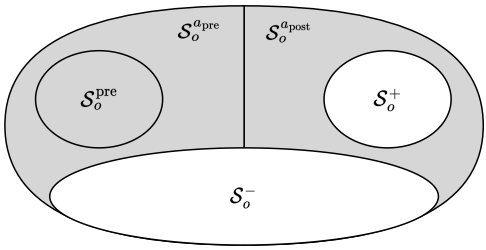


Figure 4.1: Euler diagram to visualise the relationship of different object state subsets.

We further assume that the object transfer process can either succeed or fail. We therefore define that the existence of two predicates, $P_+(s)$ and $P_-(s)$, that can evaluate if the object is either in a success or a failure state.

$$\mathcal{S}_o^+ = \{s \in \mathcal{S}_o^{a_{\text{post}}} : P^+(s)\} \tag{4.5}$$

$$\mathcal{S}_o^- = \{s \in \mathcal{S}_o : P^-(s)\} \tag{4.6}$$

Note that a success state always implies that the ownership has been changed to a_{post} while failure states can be reached even before the ownership changed. We further define the set of all non terminal state as $\mathcal{S}_o^{'}$, which is visualized by the grey area in Figure 4.1.

$$\mathcal{S}_o^{'} = \mathcal{S}_o \setminus (\mathcal{S}_o^+ \cup \mathcal{S}_o^-) \tag{4.7}$$

During the object transfer process, the state of the object o changes over time. The function $s(t)$ represents this temporal relationship. We define $s(t)$ in the closed time interval between t_0 and t_{end}.

$$s(t) : [t_0, t_{end}] \rightarrow \mathcal{S}_o \quad \text{with } s(t_0) \in \mathcal{S}_o^{\text{pre}} \tag{4.8}$$

We define t_{end} as the time at which the object enters a success or a failure state for the first time.

$$t_{end} = \min\{t : s(t) \in (\mathcal{S}_o^+ \cup \mathcal{S}_o^-)\} \tag{4.9}$$

In addition, we define a state transition function u. This function describes the state change of o during the object transfer. The resulting state depends on factors which, for the general case, are combined in the set $\mathcal{U}$. These factors may include parameters such as control signals as well as uncertain and unmeasured parameters, e.g. environmental factors affecting the transfer process.

$$u : \mathcal{S}_o^{'} \times \mathcal{U} \rightarrow \mathcal{S}_o \tag{4.10}$$

If we discretise time, we can interpret the object transfer process as a discrete sequence of states in the interval $[t_0, t_{end}]$ as follows:

$$s[t_i] = \begin{cases} s(t_0) & \text{if } t_i = t_0 \\ u(s[t_{i-1}]) & \text{if } t_i \leq t_{end} \end{cases} \tag{4.11}$$

Figure 4.2 shows a schematic representation of the object transfer process of an object o. At time t_0, the object is in a valid pre-transfer state and is moved according to its state transition function u. Between t_1 and t_2 the possession of the object changes to a_{post}. At time t_{end}, the object reaches a terminal state which is evaluated as belonging to the success states. In the application study in Chapter 6, we will use digital twins to estimate $p^+(s[t_0])$, the probability for an object in a given pre-transfer state to complete the transfer process in a state $s \in \mathcal{S}_o^+$.

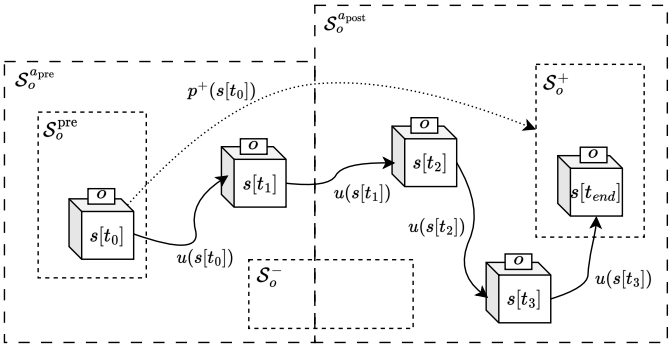
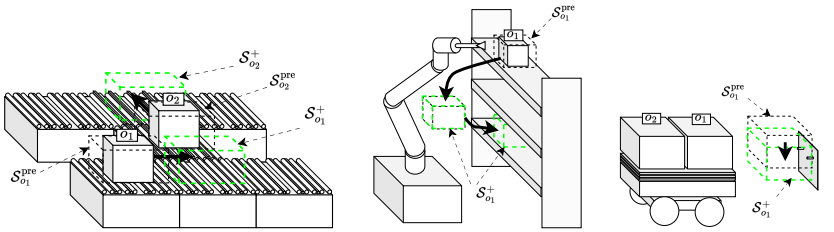


Figure 4.2: Schematic representation of an exemplary object transfer process.

4.2 Examples

In this section, we examine three different OTS examples which can described by the model proposed in Section 4.1. These include:



(a) Decentralised Conveyor Grid **(b)** Pick & Place Manipulator **(c)** Urban Delivery Robot

Figure 4.3: Exemplary object transfer systems scenarios.

- A decentralised conveyor grid, see Figure 4.3a,
- a pick & place manipulator, see Figure 4.3b,
- and an urban delivery robot 4.3c.

Figure 4.3a shows a decentralised controlled conveyor grid (Mayer and Furmans 2010). After an initial object identification by one of the conveyor modules, a route to the final destination is reserved and the object is gradually transferred to its destination module. In terms of our proposed description model, an OTS is initiated once before each transfer between two successive modules. Each module has a simple sensor setup based on an RFID reader for object identification and light barriers on all outer edges, which are used to detect the displacement of incoming and outgoing objects. In a valid pre-transfer state, the object ID is readable and all of the module's light barriers are uninterrupted ($s[t_0] \in \mathcal{S}_o^{\text{pre}}$). During a transfer, the drives of both involved conveyor modules are activated. When the object's ID is readable on the second module and the light barriers of the second module's light barriers are not interrupted, the object has reached a success state and the transfer is complete ($s[t_{end}] \in \mathcal{S}_o^{+}$).

Figure 4.3b shows a manipulator performing pick & place operations to rearrange objects on a shelf. According to our description model, a pick & place operation is a concatenation of two OTS. In the first transfer, the manipulator detects and identifies the object and validates whether its state is a valid pre-transfer state.

After grasping and lifting the object, the ownership of the object changes from the shelf (a_{pre}) to the manipulator (a_{post}). The success state of the first transfer becomes the pre-transfer state of the second transfer. If the object is successfully placed in the designated shelf space, the transfer is complete.

The last example in Figure 4.3c shows an urban delivery robot transporting two boxes filled with parcels. The object transfer involves clipping the box to a fixture by means of a vertical downward lift. We could define valid pre-transfer states when the mobile robot successfully docked to the bracket with an acceptable docking error. After the box is lowered and physical contact is lost, the ownership of the object is changed a successful object state is reached.

4.3 Chapter Conclusion

In this chapter we have presented a formal description model for the simplest form of an *Object Transfer System* (OTS). In this form, an OTS consists of three assets, an asset that possesses the object to be transferred, an asset that is intended to possess the object after the transfer process, and the object to be transferred itself. The core of the model evolves around a state description of the object to be transferred. During a transfer, the state of the object changes over time. We therefore model the object transfer process as a state transition function. We define several important sets of states that are used to control the object transfer process including the pre-transfer states $\mathcal{S}_o^{\text{pre}}$, the success states $\mathcal{S}_o^+$, the failure states $\mathcal{S}_o^-$ and the non-terminal states $\mathcal{S}_o'$. In addition, we have provided three examples to show that the model is applicable to different object transfer scenarios.

5 Implementation of an Object Transfer System in a Remanufacturing Demonstration Environment

The content of this chapter extends upon sections of a working paper titled "A Study on the Predictive Capabilities of Digital Twins for Object Transfers in Fluid Automated Remanufacturing" (Klein and Furmans 2025). The author of this thesis was responsible for the conceptualisation, methodology, software programming, validation, formal analysis, writing, and visualisation of the research presented in this chapter.

This chapter presents a real-world implementation of an *Object Transfer System* (OTS) that follows the description model introduced in Chapter 4 and at the same time, represents an implementation of the DT architecture introduced in Chapter 3. The OTS was developed as part of the *AgiProbot* project (Agile Production System using Mobile, Learning Robots with Multisensor Technology for Uncertain Product Specifications), funded by the Carl Zeiss Foundation. The interdisciplinary project was carried out by a collaboration of different institutes at the Karlsruhe Institute of Technology (KIT) from 2019 to 2024. In general, the *AgiProbot* project aims to develop new methods to increase the level of automation in remanufacturing, with a focus on disassembly processes (Lanza et al. 2022). The methods are physically demonstrated using automotive servomotors, which are commonly used as seat adjuster or windshield wiper motors, and starter motors, which are currently remanufactured by Bosch as

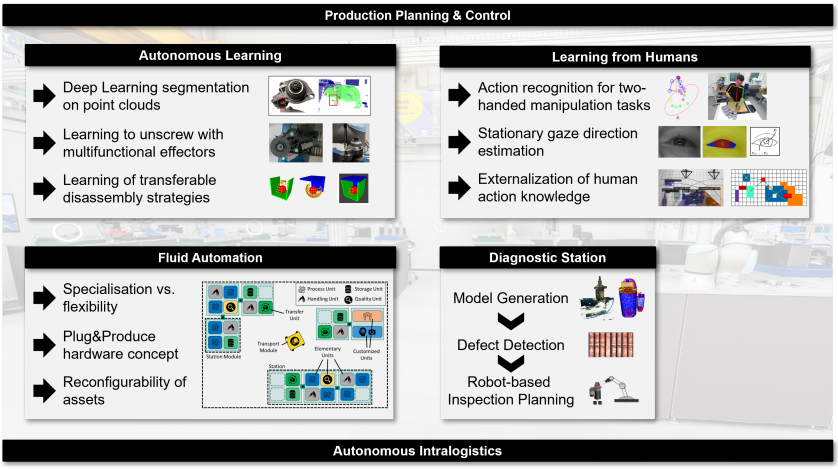
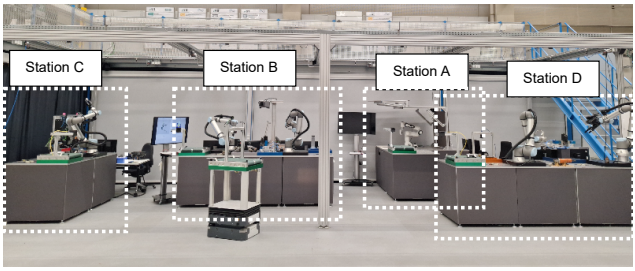


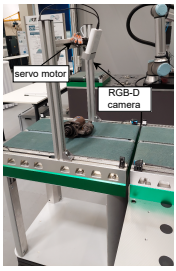
Figure 5.1: Main research areas of the AgiProbot project.

part of the "Bosch eXchange program" (Tolio et al. 2017). Figure 5.1 gives an overview on the project's main research areas, product core diagnostics (Kaiser et al. 2022), learning from human demonstration (Dreher and Asfour 2022, Leven and Heizmann 2022), autonomous learning for robotic systems (Wu et al. 2023, Mangold et al. 2022, Cebulla et al. 2023), plug-and-produce production systems (Wurster et al. 2021), production planning and control (Wurster et al. 2022) as well as autonomous intralogistics (Klein et al. 2021).

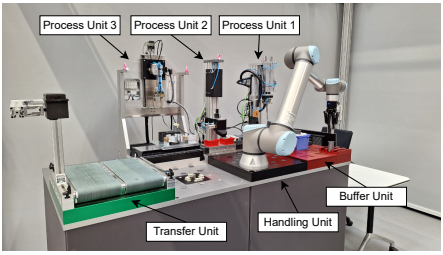
The chapter is structured as follows. Section 5.1 gives a detailed description of the implemented physical assets. Section 5.2 continues with the implemented logical assets including the virtual models. Section 5.3 gives a complete overview of the implemented CPS. Section 5.4 describes how the presented assets form an OTS and the process flow of a transfer execution. After discussing challenges of the implementation in Section 5.5, we conclude the chapter in Section 5.6. Note that the object transfer process, in particular the incorporation of the virtual models, is examined in the following application study in Chapter 6.



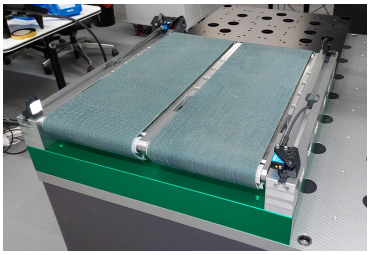
(a) Total view of the *AgiProbot* demonstration production system including four stations and the transport module, see Klein et al. 2021.



(b) Transport Module, see (Klein et al. 2021).



(c) The *Fluid Automation Station* (Station C) of the *AgiProbot* project: A realisation of the *Fluid Automation Framework*, see Wurster et al. (2021).



(d) Transfer Unit: Two separated controlled conveyor belts with two light barriers, see (Klein et al. 2021).

Figure 5.2: The *AgiProbot* production system, see (Wurster et al. 2021, Klein et al. 2021).

5.1 Physical Assets $\mathcal{A}^P$

The AgiProbot production system is an implementation of the *Fluid Automation Framework*, which defines a highly reconfigurable and service-oriented production system (Wurster et al. 2021). At the top level, the system is divided into *Stations*, each of which consists of at least one *Station Module*. Station modules provide standardised interfaces for power, communication and compressed air for up to four so called *Elementary Units*. In the *Fluid Automation Framework*, elementary units are the smallest described entities, each of which provides specific capabilities as services to the overall system (Wurster et al. 2022). The fluid automation design allows for rapid reconfiguration, which is part of the solution

set to cope with the inherent uncertainties of remanufacturing systems. For more information on the *Fluid Automation Framework* and its use in production planning and control, please refer to (Wurster et al. 2021, Wurster et al. 2022).

The AgiProbot demonstration system comprises a total of four stations each of which is used to explore the different research areas shown in Figure 5.1. Figure 5.2a shows all four stations on the common shop floor. A close-up view of *Station C* is shown in Figure 5.2c. This station best illustrates the implementation of the *Fluid Automation Framework*. The station consists of two station modules with a total of six elementary units currently mounted. Each elementary unit is a self-contained system that can be easily mounted on a station module thanks to standardised hardware interfaces. Figure 5.2d and Figure 5.2b show the two main asset types that are used to form the object transfer systems, namely the *Transfer Unit* and the *Transport Module*. Both assets were first introduced in Klein et al. 2021 and are explained in detail in the following subsections. In addition, we describe the set of starter motor cores, which we consider as the set of physical objects to be transferred.

5.1.1 Transfer Units $\mathcal{TU}^P$

Transfer units are elementary units and are used as stationary transfer points. Given the formal model of an OTS introduced in Section 4.1, transfer units can act both as a_{post} when an object is being transferred to a station module and as a_{pre} when an object is being collected by the transport module. The AgiProbot demonstration system includes a total of four real-world transfer units, one mounted on each station.

$$\mathcal{TU}^P = \{TU_A^P, TU_B^P, TU_C^P, TU_D^P\} \tag{5.1}$$

In the rest of this thesis, we use the superscript ^{P} to refer to implemented physical assets. Figure 5.3 shows a view on the layered asset implementation. On the lowest layer, a transfer unit consists of a single-board computer which controls two independently actuated conveyor belts. The HAL includes interfaces to

set and read the conveyor speed and read the light sensors, which create light barriers on both outer edges of the conveyor belts. We classify the transfer unit as a mixed type **D/E** physical asset. Compared to a full type **E** asset, the transfer unit is not able to perform all aspects of its state synchronisation. For example, while the current speed of the conveyors is updated by the unit itself, the transfer unit is not able to observe if any object is placed on its conveyor belts. This part of the state synchronisation must be performed by another asset, as the transfer unit lacks the necessary sensors to perform this function independently. The functionality to be updated externally is provided on the application layer. The external communication is implemented as an OPC UA server that uses the structure of an AAS as its information model.

5.1.2 Transport Modules $\mathcal{TM}^P$

The transport module implemented in the AgiProbot project is an enhanced mobile robot, which is used to connect the different transfer units mounted on the stations in a flexible way. A single transport module has been implemented for the AgiProbot demonstration system. We refer to this transport module as TM_1^P.

$$\mathcal{TM}^P = \{TM_1^P\} \tag{5.2}$$

Figure 5.4 shows the layered asset implementation of the transport module. Starting from the bottom layer, the transport module consists of three main subsystems, a laser-based AMR as a mobile platform, a transfer unit as a load handling module and a vision system which is used to detect and track objects during object transfers. The vision system combines a Microsoft Azure Kinect RGB-D camera and a small servo motor that allows the camera to be rotated to change its view. The HAL provides interfaces to the various sensors and actuators. The control layer includes software modules to solve specific tasks, such as the object detector and tracker or AMR-related modules for localisation and path planning. The application layer consists of modules for setting and executing drives to a target pose on a pre-mapped occupancy grid, docking to a

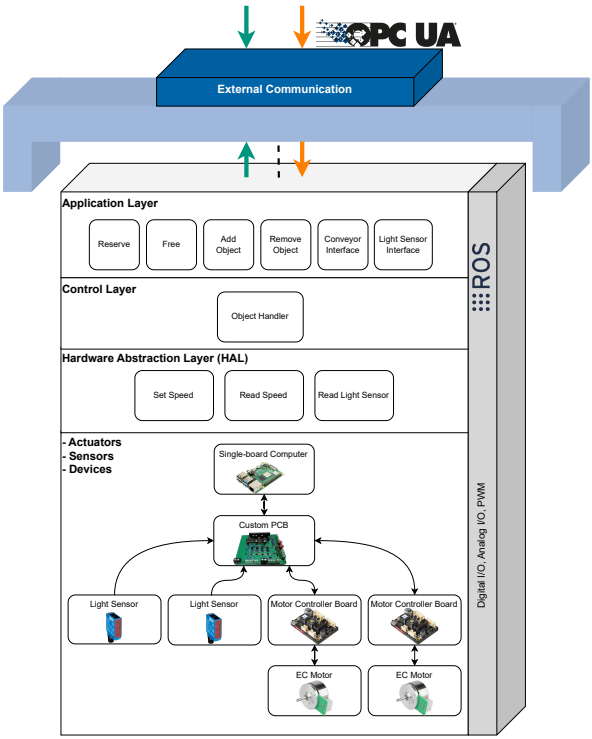


Figure 5.3: A transfer unit, a mixed type **D/E** physical asset.

transfer unit or initiating an object transfer process. The various capabilities of the transport module and its interaction with a transfer unit and its environment are discussed in detail in Section 5.4 as part of the object transfer process. We classify the transport module as an asset of type **E**.

5.1.3 Objects $\mathcal{O}^P$

As described in the formal model in Section 4.1, an OTS consists of three involved assets. Besides the transfer unit and the transport module as the

possessing assets, we need to define the objects to be transferred. Although the described OTS is capable of transferring a wide range of different objects, we consider a fixed set of 37 real-world starter motor cores as the set of objects to be transferred.

$$\mathcal{O}^P = \{o_1^P, ..., o_{37}^P\} \tag{5.3}$$

Each core considered is not only unique in its degree of wear, but also originates from a different product variant. Besides being unique, the starter motors share the same product family, which makes them similar in terms of their overall design scheme. Figure 5.5 shows images of the set $\mathcal{O}^P$. The influence of the object's shape on the success probability of the object transfer is further explored in the DT application study in Chapter 6. In the CPS, the objects are implemented

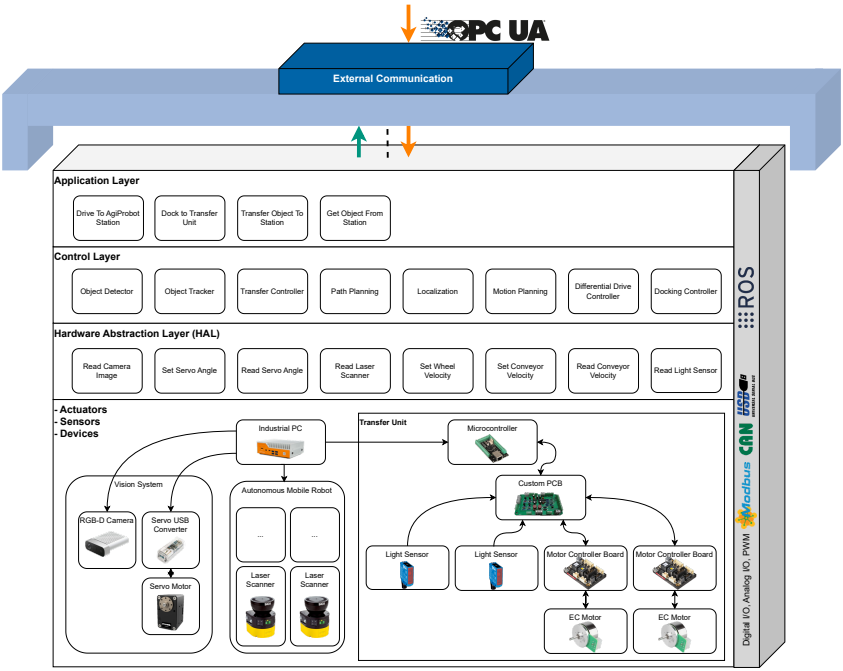


Figure 5.4: A transport module, a type **E** physical asset.

as assets of type **B**, which are incapable of internal state synchronisation and have no external control interfaces.

Figure 5.5: Set of unique starter motor cores $\mathcal{O}^P$.

5.2 Logical Assets $\mathcal{A}^L$

Besides the physical assets, we implemented several logical assets that provide additional services in the CPS:

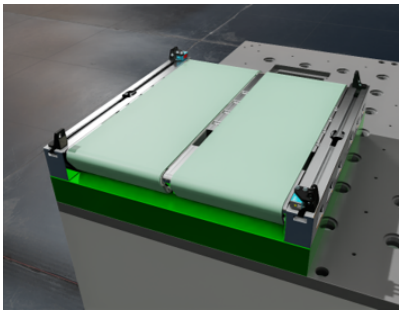
- The *Transport Module System*, previously mentioned in Klein et al. 2021, represents the global interface to the production control. It allows the placement of transport operations, which can be distributed to the transport modules with the help of task allocation procedures.
- *PropS* is a logical asset which is used to fuse distributed pose information to enable collaborative localisation. The system structure and a proof of concept implementation were presented in Klein et al. 2022. Based on

a unified AAS submodel, we demonstrated how *PropS* can be used to reason about the docking pose of an AMR in a dynamically changing environment. Although the docking pose is unknown to the robot itself, it can be inferred from multiple cameras distributed across multiple assets.

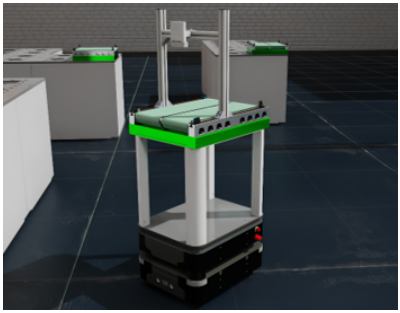
- For the application study presented in Chapter 6, we implemented an *Experiment Manager*. This asset manages the planning and execution of object transfer experiments for both physical assets and virtual models. This includes creating of virtual environments, instantiating of virtual models, interacting with the assets involved, storing the results in a database and handling of tasks related to post-processing of the results.

5.2.1 Virtual Models $\mathcal{A}^V$

The virtual models are the core of the digital twin implementation. Instances of virtual models are created and used in a virtual environment, which provides the framework for simulating space and time. As a virtual environment we use NVIDIA Isaac Sim, a popular multi-body simulator that uses NVIDIA's physics engine PhysX (NVIDIA Cooperation 2024). For each physical asset introduced in Section 5.1, we consider a corresponding virtual model. We use the superscript ^{V} to refer to the virtual models. Figure 5.6a shows an instance of



(a) Instance of a virtual transfer unit TU^V



(b) Instance of a virtual transport module TM^V

Figure 5.6: Virtual models in an NVIDIA Isaac simulation environment.

the virtual transfer unit (TU^V) and Figure 5.6b shows an instance of the virtual transport module (TM^V). For details on the modeling procedure, we refer to (Klein and Furmans 2023). When implemented as an asset in the CPS, the virtual models reuse most of the software models shown in the corresponding figures in Section 5.1. As the hardware layer is completely replaced by the virtual sensors and actuators, the components on the HAL are adapted in terms of their interface to the lower layer. The control layer remains unchanged compared to the physical assets. In addition to the virtual transfer units and transport modules, we consider a set of virtual objects models $\mathcal{O}^V$ which matches the physical object set $\mathcal{O}^P$. Figure 5.7 shows the virtual object models. For the DT application study, the virtual object models are simulated as simple rigid body with uniformly distributed mass and a high resolution STL model as a collision shape. The STL models were generated by scanning $\mathcal{O}^P$ with a ZEISS T-Scan industrial laser at the AgiProbot diagnostic station.

Compared to the physical assets, the number of virtual model instances is easily scalable, limited only by the available computing capacity. The virtual models can be instantiated in a single virtual environment or in multiple virtual environments across a distributed system of computing nodes. Given the forecasting function of a DT, this allows to parallelise the prediction to achieve significant results in a shorter period of time.

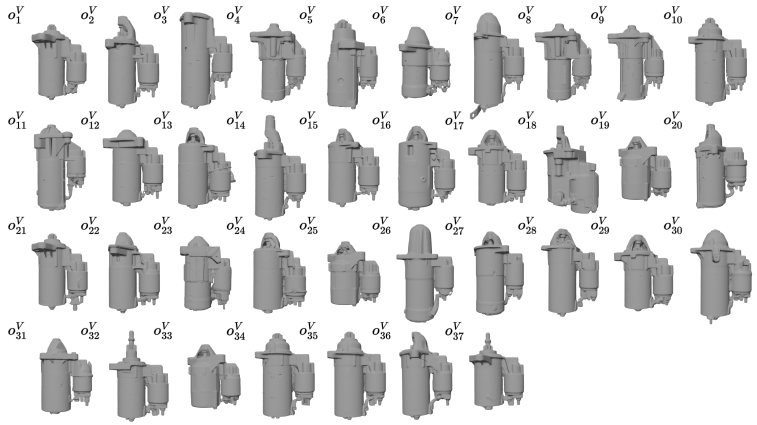


Figure 5.7: Set of virtual starter motor cores $\mathcal{O}^V$.

5.3 The Cyber-Physical System

Based on the architecture introduced in Chapter 3, Figure 5.8 shows the implemented CPS to enable DT-based object transfers as part of the AgiProbot demonstration system. It includes the physical assets $\mathcal{TU}^P$, $\mathcal{TM}^P$ and $\mathcal{O}^P$, three logical assets, namely the *Transport Module System*, *PropS* and the *Experiment Manager* and a varying number of virtual models, $\mathcal{TU}^V$, $\mathcal{TM}^V$ and $\mathcal{O}^V$. All assets have a virtual representation which data structure is based on the AAS specification and use OPC UA as their external communication interface. In principle, all assets can establish communication channels to all other assets. However, in practice, only a small subset of communication channels is needed to realise an object transfer, for the physical assets as well as the virtual models. The transport module is the main actor during the object transfer process. It uses the control interface of the transfer units and updates information on the objects to be transferred. Please note that the illustrated communication channels in Figure 5.8 are not complete. *PropS*, e.g., establishes connections to all virtual representations to update its internal pose graph. The interaction of the

assets with the knowledge graph is not explored as part of this thesis, but will be explored in future work.

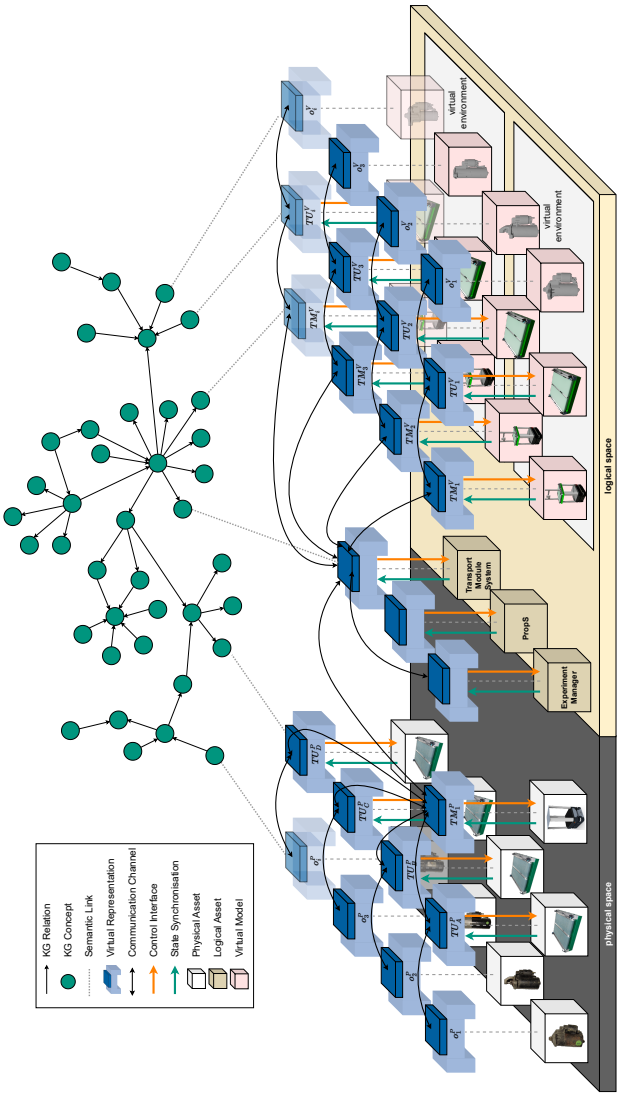


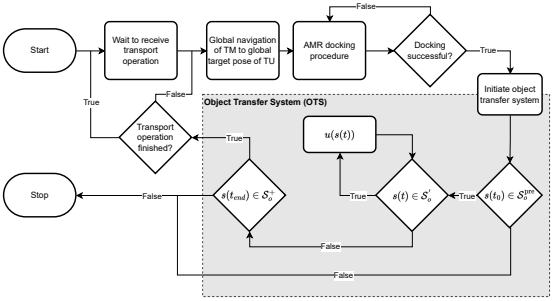
Figure 5.8: Implemented CPS, part of the *AgiProbot* demonstration system.

5.4 Object Transfer System

Through cooperation, the introduced assets enable the material flow between the four stations of the AgiProbot demonstration system by executing transport operations. We define a transport operation as a transport task which has already been matched and scheduled on a specific resource, see (Pfrommer et al. 2022). Figure 5.9a shows a flowchart of the execution of a transport operation. The transport operation, which is initially received by the *Transport Module System* is passed to the idle transport module. As explained in (Zhou et al. 2023), the transport module first performs a global navigation process to the target pose of the transfer unit, which is extracted from a pre-generated hypermap. Once TM_1^P reaches the coarse target pose, the navigation method is switched to a custom laser-based method that uses the cuboid shape of the station modules as a reference to estimate the docking pose. Figure 5.9b shows a specific scenario in which TM_1^P successfully docked at TU_B^P. Next, the OTS is being initiated by TM_1^P who establishes a connection to the virtual representation of a transfer unit TU^P. The transport operation can contain one of two different transfer sequences:

- o^P is being transferred from TM_1^P to TU^P, $a_{pre} = TM_1^P, a_{post} = TU^P$
- o^P is being transferred from TU^P to TM_1^P, $a_{pre} = TU^P, a_{post} = TM_1^P$

Before the actual transfer process begins, TM_1^P uses its vision system to perform a full scan of the surface of its own conveyors and those of the docked TU^P to match detected objects with the information available on the virtual representations. The OTS is successfully formed when all requirements for starting the execution of an object transfer are met. This includes, e.g., that no other object is already occupying the target conveyor belt. Next, TM_1^P evaluates whether the object is in a valid pre-transfer state. If the state is valid, the transfer process loop starts by constantly monitoring if the object state s is still a non-terminal state. Once o^P reaches either a success or failure state, the OTS gets resolved.



(a) Flowchart of the transport operation execution.

(b) Docked transport module.

Figure 5.9: The transport module (TM_1^P) after a successful docking.

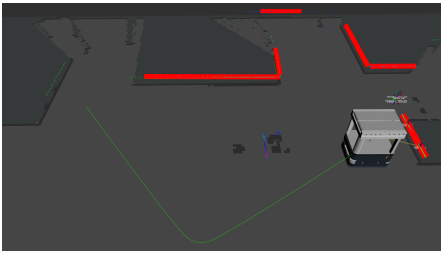
If the transfer was successful, TM_1^P can continue with further sub-operations or TM_1^P returns to its idle state and waits for further transport operations.

The following subsections provide additional implementation details. First, in Section 5.4.1, we present details of the global navigation and docking process. Section 5.4.2 describes the system-dependent definition of the object states. Section 5.4.3 introduces the implemented object detector and tracker, and Section 5.4.4 provides further details on the applied visual servoing method.

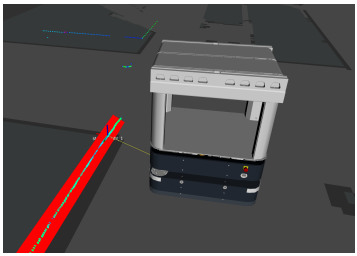
5.4.1 Global Navigation and Docking Process

During the global navigation process, TM_1^P plans a collision-free path to a preliminary docking pose in front of the target station. The generated path is adapted by the motion planning module to be able to avoid dynamic objects during the execution. Figure 5.10a shows a top view of the TM_1^P on the pre-mapped occupancy grid after generating a path to *Station C*.

Once the transport module has reached its coarse target pose, a laser-based method is used to estimate the final docking pose. The procedure takes advantage of the cuboid shape of the station modules. First, line candidates are segmented in the raw laser data, red lines in Figure 5.10a, and the closest line that matches



(a) Global navigation of TM_1^P on the *AgiProbot* demonstration shop floor. After undocking from *Station D*, an initial path (green line) is planned to the coarse docking pose of *Station C*.



(b) Laser-based pose estimation of the docking pose associated with the transfer unit, see (Zhou et al. 2023).

Figure 5.10: Navigation modes of the implemented transport module TM_1^P.

the configuration of the target station is selected. Based on the outer edge of the selected line, a docking pose is estimated. Figure 5.10b shows the matched line candidates and the estimated docking pose. For more details on the docking procedure, we refer to (Zhou et al. 2023).

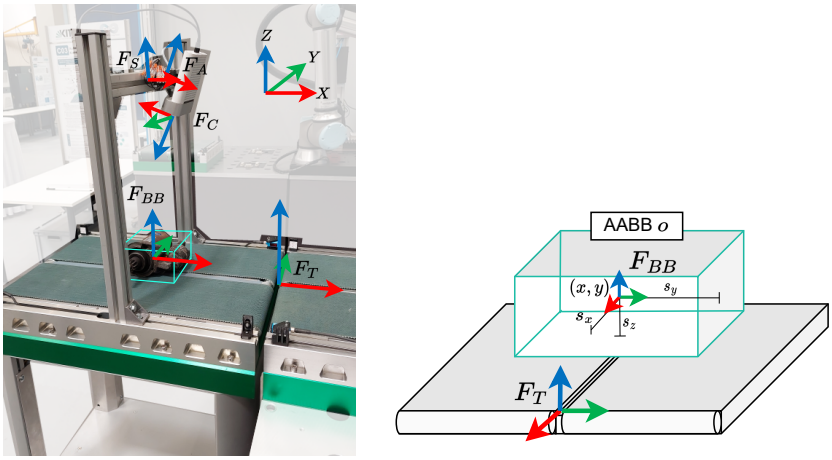
Table 5.1: Overview of the relevant coordinate frames in a formed OTS.

Frame	Description
F_T	Fixed frame defined at the front tip of TM_1^P.
F_S	Fixed frame defined at the mounting point of the servo module.
F_A	Rotatable frame defined in the axis of rotation of the servomotor.
F_C	Fixed camera frame with z-axis as optical axis.
F_{BB}	Frame at the centre of an Axis Aligned Bounding Box.
F_O	Object specific frame for the starter motor set. The positive x-axis points towards the starter pinion, the positive y-axis points in the opposite direction of the engagement relay.

5.4.2 Object States Definition

In Section 4.1 we introduced a formal description of an object transfer system. The object transfer process itself is described in terms of state changes of the considered object o^P. The object state s is generally defined by the possessing asset and additional, system-dependent components, see Equation 4.1. For the case of the object transfer system introduced in this chapter, we define $s \in \mathcal{S}_o$ as follows:

$$\mathcal{S}_o = \{(a, x, y, z, s_x, s_y, s_z, \theta, f, c, t)\} \tag{5.4}$$



(a) Overlayed coordinate frames on a docked transport module.

(b) State parameters related to an Axis Aligned Bounding Box (AABB): The AABB is fully described by its centre point (x, y), and the half extents (s_x, s_y, s_z) of the box.

Figure 5.11: Coordinate frames as listed in Table 5.1 and state parameters associated with an Axis Aligned Bounding Box (AABB).

As defined in Equation 4.1, a refers to the asset in possession of o. The component subset (x, y, z, s_x, s_y, s_z) is associated with the pose of o. In our case, the transport module uses its vision system to estimate this pose by computing an *Axis Aligned Bounding Box* (AABB) for each tracked object. Figure 5.11b

shows a schematic of a bounding box detected on the conveyor belt pair. As the bounding box is axis aligned, it does not provide any information about the orientation of the tracked object. We define θ as the yaw angle rotation of the object. Although the orientation of the object cannot be estimated during the transfer process, it is an important value for defining the pre-transfer state of an object. Specifically for the starter motor set, we assign a right-handed coordinate frame (F_O) with the positive x-axis pointing towards the starter pinion and the positive y-axis pointing in the opposite direction of the engagement relay.

We further define f, the *face* on which the object is placed on. This parameter is specifically defined for the starter motor set. During initial testing we found that all starter motors, due to their shape, can only be safely placed on either one of two faces:

$$f \in \{\text{'front'}, \text{'back'}\} \tag{5.5}$$

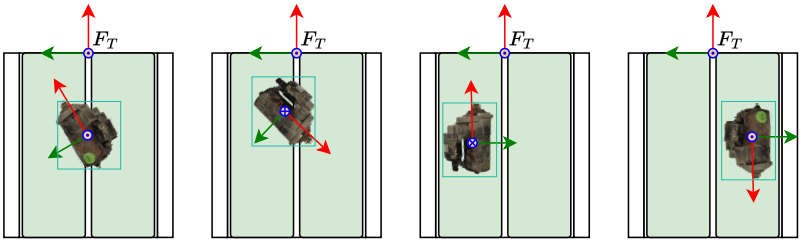
Figure 5.12a and 5.12d show the object on the 'front' face, with the engagement relay pointing to the left side with respect to the object's centre and the z-axis of the object frame pointing upwards. Figure 5.12b and Figure 5.12c show the object on the opposing 'back' face.

In addition, we define three different conveyor positions, see Figure 5.12:

$$c \in \{\text{'left'}, \text{'right'}, \text{'both'}\} \tag{5.6}$$

The conveyor parameter c is set depending on whether there is an overlap of the tracked AABB with the corresponding conveyor. c is used during the transfer process to decide which conveyor belt needs to be active. In the Figures 5.12a and 5.12b, the object occupies both conveyor belts, in Figure 5.12c, the object is placed on the left belt, and in Figure 5.12d on the right belt.

The last component of the state vector is the time t, which describes the elapsed time since the OTS was formed. Figure 5.12 shows four different examples of object states. In each sub-figure, o_1^P is located on a transport module TM_1^P, but in a different pose (x, y, θ). Note that even though the object is identical, the dimensions of the AABB vary due to the different orientations of the object.



(a)
$$\begin{pmatrix} TM_1^P \\ -0.3m \\ 0.0m \\ 0.11m \\ 0.10m \\ 0.09m \\ 30° \\ \text{'front'} \\ \text{'both'} \\ t_0 \end{pmatrix}$$

(b)
$$\begin{pmatrix} TM_1^P \\ -0.2m \\ 0.02m \\ 0.11m \\ 0.10m \\ 0.07m \\ -135° \\ \text{'back'} \\ \text{'both'} \\ t_0 \end{pmatrix}$$

(c)
$$\begin{pmatrix} TM_1^P \\ -0.325m \\ 0.094m \\ 0.13m \\ 0.08m \\ 0.07m \\ 0° \\ \text{'back'} \\ \text{'left'} \\ t_0 \end{pmatrix}$$

(d)
$$\begin{pmatrix} TM_1^P \\ -0.325m \\ -0.094m \\ 0.13m \\ 0.08m \\ 0.09m \\ 180° \\ \text{'front'} \\ \text{'right'} \\ t_0 \end{pmatrix}$$

Figure 5.12: Examples of different pre-transfer states: Each schematic shows the top view of a transport module with a single object located on its conveyor belts. The green square visualises the AABB.

As introduced in Section 4.1, we further define the state subsets $\mathcal{S}_o^{\text{pre}}$, $\mathcal{S}_o^+$, $\mathcal{S}_o^-$ and $\mathcal{S}_o^{'}$. Before an object transfer system process is initated, the object must be in a valid pre-transfer state $s \in \mathcal{S}_o^{\text{pre}}$:

$$\mathcal{S}_o^{\text{pre}} = \left\{ s \in \mathcal{S}_o \;\middle|\; \begin{pmatrix} a = a_{\text{pre}} \\ t = t_0 \\ (x - s_x \geq -0.65m) \wedge (x + s_x \leq 0.0m) \\ (y - s_y \geq -0.2m) \wedge (y + s_y \leq 0.2m) \end{pmatrix} \right\} \tag{5.7}$$

We have defined four conditions that all must be met to limit the state space to the pre-transfer states. First, a_{pre} must be in possession of o. Since the transfer process has not yet started, t is still set to t_0. The remaining two conditions make sure that AABB does not overlap with the area outside the conveyor belts of a_{pre}. All exemplary states visualised in Figure 5.12 are valid pre-transfer states.

$\mathcal{S}_o^+$ describes the set of states that an object is in when a transfer process has been completed successfully

$$\mathcal{S}_o^+ = \left\{ s \in \mathcal{S}_o \;\middle|\; \begin{pmatrix} a = a_{\text{post}} \\ t < t_{max} \\ x > 0.2m \\ (x - s_x \geq 0.0m) \wedge (x + s_x \leq 0.5m) \\ (y - s_y \geq -0.2m) \wedge (y + s_y \leq 0.2m) \end{pmatrix} \right\} \tag{5.8}$$

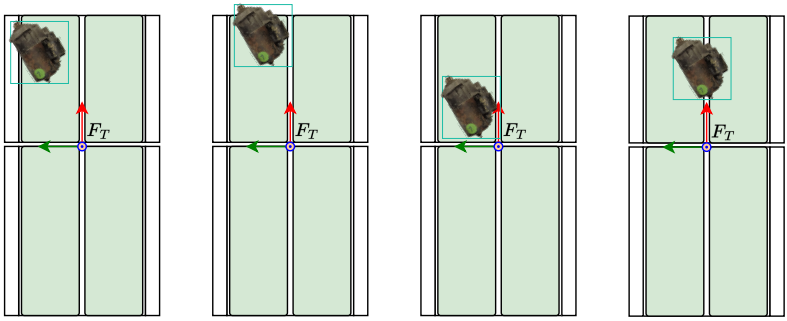
We have defined a total of five conditions that must all be met for an object state to be in $\mathcal{S}_o^+$. First, the possession must have changed to a_{post}. For our object transfer system, we have defined a timeout time t_{max} until the transfer process must be completed. For a transfer to be successful, the object must be safely moved to a_{post}. To ensure this, we set a threshold value for x which is the minimum translation of the object along the direction of the transfer unit. $0.2m$ was evaluated as a suitable value, based on the dimensions of the transfer unit and the average length of the considered starter motor cores.

The remaining two conditions make sure that AABB does only overlap with the belts of a_{pre} and not with any area outside of the conveyor belts.

In contrast, the set $\mathcal{S}_o^-$ holds all states that are associated with a failed transfer process.

$$\mathcal{S}_o^- = \left\{ s \in \mathcal{S}_o \;\middle|\; \begin{pmatrix} (t > t_{max}) \vee ((x - s_x < -0.65m) \wedge (x + s_x > 0.5m)) \\ (t > t_{max}) \vee ((y - s_y < -0.2m) \wedge (y + s_y > 0.2m)) \end{pmatrix} \right\} \tag{5.9}$$

An object is in a failure state if either the timeout t_{max} has passed, or the AABB of the tracked object does overlap with the area outside of any of the conveyor belts. Figure 5.13 shows four object state examples after the completion of the object transfer process. The AABBs of objects in Figure 5.13a and 5.13b both overlap with the area outside the conveyors. In Figure 5.13c, the object has not surpassed the threshold value of $x > 0.2m$. Figure 5.13d shows a successful



(a) $s(t_{end}) \in \mathcal{S}_o^-$ **(b)** $s(t_{end}) \in \mathcal{S}_o^-$ **(c)** $s(t_{end}) \in \mathcal{S}_o^-$ **(d)** $s(t_{end}) \in \mathcal{S}_o^+$

Figure 5.13: Examples of success and failure states.

object state. Finally, $\mathcal{S}_o^{'}$ describes all non terminal states. The object transfer process continues $(u(s(t)))$ as long as the object is in a state $s \in \mathcal{S}_o^{'}$.

$$\mathcal{S}_o^{'} = \mathcal{S}_o \setminus (\mathcal{S}_o^+ \cup \mathcal{S}_o^-) \tag{5.10}$$

For our use case, we interpret the OTS as a binary function returning a boolean value depending if the transfer was successful or not:

$$OTS(o, a_{\text{pre}}, a_{\text{post}}) = \begin{cases} 1 & \text{if } s(t_{end}) \in \mathcal{S}_o^+ \\ 0 & \text{if } s(t_{end}) \in \mathcal{S}_o^- \end{cases} \tag{5.11}$$

In the following, the detection and tracking method is presented. This method outputs the AABBs.

5.4.3 Detector and Tracker

The actuated vision system is used to detect and track objects during an object transfer process. The detection pipeline is shown in Figure 5.14. The pipeline

runs continuously and processes each captured point cloud by the RGB-D camera. During the preprocessing steps, the raw point cloud is transformed into the fixed frame F_T after an initial removal of invalid points. Next, the dimensions of the OTS workspace are calculated. The workspace describes the space in which objects are expected. If TM_1^P is not docked to any TU^P, the workspace includes only the two conveyor belts of TM_1^P. If TM_1^P is already docked, the workspace is extended to include the conveyor belts of TU^P. The point cloud is filtered based on this workspace. To reduce the computational load in the subsequent steps, we apply a voxel grid filter to downsample the point cloud. Since the points corresponding to the conveyor belts can be represented as a plane, we use plane segmentation to eliminate them. A widely used approach for plane segmentation in point clouds is the RANdom SAmple Consensus (RANSAC) algorithm, which has been successfully applied in similar contexts, as demonstrated in (Navarro et al. 2014, Bras and Neto 2018). Next, we use a statistical outlier algorithm based on k-nearest neighbours to remove any remaining outliers. For the classification step, we first perform a euclidean cluster extraction to distinguish between different objects placed on the conveyors. Finally, for each cluster, we compute an axis-aligned bounding box (AABB).

The tracker uses the stream of states for $\mathcal{O}_{detected}$ to track the movement of individual objects. The tracker tries to match each element of $\mathcal{O}_{detected}$ with its stored set $\mathcal{O}_{tracked}$. The matchmaking algorithm uses the centre point of the bounding boxes. In some cases, the object may not be detected or the detection may be inaccurate. If an object is not detected in several consecutive frames while the conveyor belt is moving, the matchmaking between $\mathcal{O}_{detected}$ and $\mathcal{O}_{tracked}$ becomes difficult. To avoid this scenario, we use a Kalman filter to estimate the position, taking into account the speed of the conveyor belts.

The proposed method allows the detection and tracking of arbitrary objects. However, the method does not allow any form of object recognition. It only detects the existence of an object, not its identity. The ability to detect and track multiple objects simultaneously allows the transport module to buffer multiple objects and to chain transport operations. Combined with the design decision to

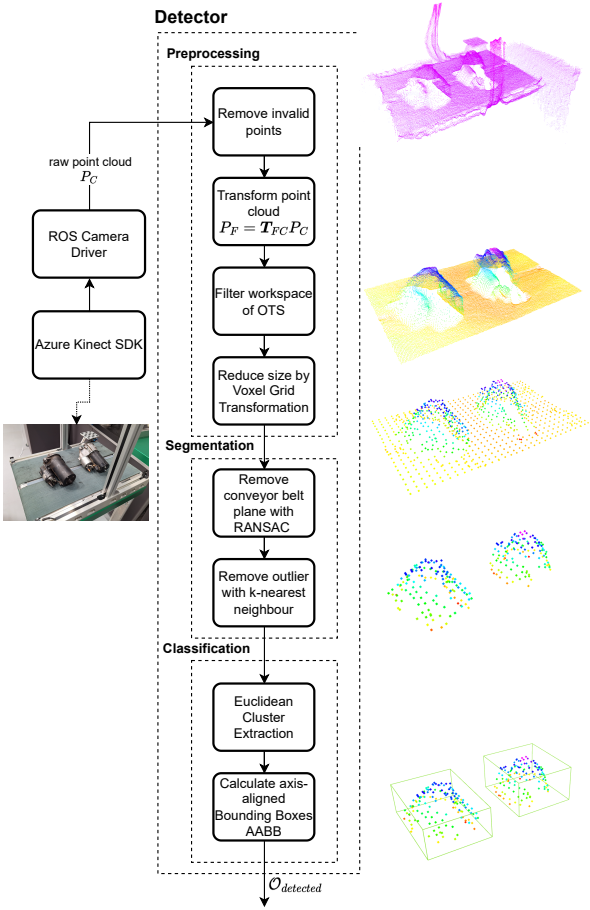


Figure 5.14: Implemented detection pipeline.

use two separate conveyor belts for the transfer unit, objects placed on one belt can be transferred without affecting other objects on the other belt.

5.4.4 Visual Servoing

During the object transfer process, the vision system must ensure that the object is always visible to the camera sensor. This is achieved by dynamically adjusting the rotation of the camera to keep the optical axis of the camera pointing at the centre of the object as the conveyor belts move the objects linearly. This can be described as a case of visual servoing (Janabi-Sharifi et al. 2011). Given the geometric constraints of our OTS, the task is to determine the servo angle α to achieve this alignment. Since the camera can only rotate around one axis, the 3D problem can be simplified to a 2D one, as shown in Figure 5.15a. For the camera (F_C) to focus the object (F_O), two conditions must be met, see Figure 5.15a:

- The x-component of the homogeneous vector $\vec{\boldsymbol{d}}$ from F_C to F_{BB} expressed in F_C is 0.

$$ {}_C\vec{\boldsymbol{d}}_{CBB} = \begin{bmatrix} 0 \\ d_z \\ 1 \end{bmatrix} \tag{5.12} $$

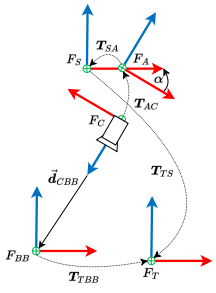
- When transformed to F_{BB}, $\vec{\boldsymbol{d}}$ is zero.

$$ {}_{BB}\vec{\boldsymbol{d}}_{CBB} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.13} $$

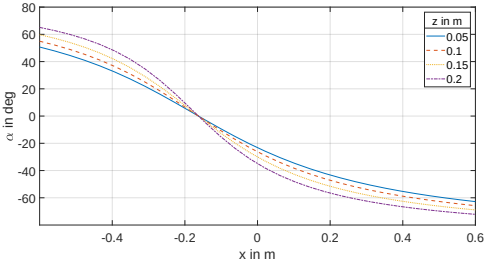
The transformation can be expressed by the following transformation chain:

$$ {}_{BB}\vec{\boldsymbol{d}}_{CBB} = \boldsymbol{T}_{TO}^{-1}\, \boldsymbol{T}_{TS}\, \boldsymbol{T}_{SA}(\alpha)\, \boldsymbol{T}_{AC}\, {}_C\vec{\boldsymbol{d}}_{CBB} \tag{5.14} $$

We can solve this equation for the two unknown values (d_z, α). Figure 5.15 shows results for the servo angle α given different object positions (x, z). At $a = 0°$, the camera's optical axis is pointing straight downwards. One can observe, that at this angle, the function is independent of the value of z.



(a) Simplified 2D sketch of the coordinate frames shown in Figure 5.11a



(b) Results for solving Equation 5.14: Given the 2D object (x, z) in F_T, the plot shows the servo angle α that needs be set for the camera to be focused on the object.

Figure 5.15: Visual servoing results.

5.5 Challenges

In this section, we summarise some of the challenges related to the assets and the methods used to implement the object transfer system.

Global Navigation

As shown in figure 5.10a, the target pose of the global navigation is defined at a certain distance in front of the transfer unit. As the mobile platform relies on non-holonomic kinematics based on a differential drive, the platform is not free to move in any direction. It is therefore important to ensure that the coarse target pose of the platform after the global navigation process completed is not too close to the station module. Otherwise there may be no feasible path to the docking pose given the curvature constraints.

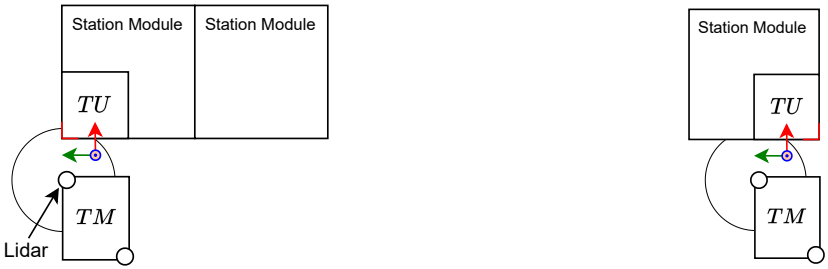
Docking

As the docking method used is based solely on laser data, there is no need for any infrastructure intervention, such as the placement of fiducial markers. This

increases the flexibility and reduces the effort required to reconfigure the system. However, the accuracy of the method varies depending on the station module configurations and the location of the transfer unit. Figure 5.16 shows two different configurations. In Figure 5.16a, the transfer unit is mounted at the front left mounting position of a station module. In this configuration, the outer edge, which is the key feature of the pose estimator, is aligned with the laser scanner, which is located at the front left corner of the transport module (symbolised by circle). In this case, the edge detection is most accurate, which directly leads to increased accuracy in the docking pose estimate. In Figure 5.16b, the transfer unit is mounted at the right mounting position of the station module. Due to the relative translation of the laser scanner to the edge of the station module, the edge detection is less accurate, resulting in a less accurate docking pose estimation. The docking error has a significant impact on the transferability of certain objects which is investigated as part of the DT application study in Chapter 6.

Detection and Tracking

When TM_1^P is docked to TU^P, the designated workspace of the OTS combines both the conveyor belt pairs of TM_1^P and TU^P. Due to its narrow field of view, the depth camera is unable to capture the entire workspace in a single frame.

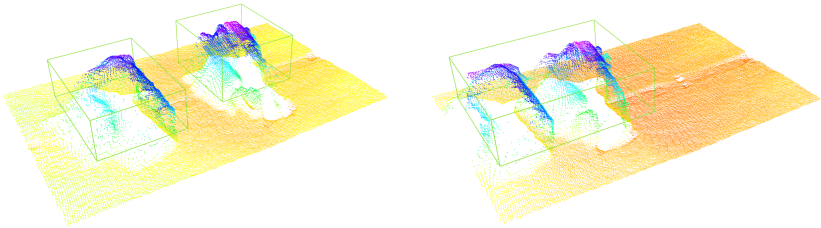


(a) Configuration 1

(b) Configuration 2

Figure 5.16: Schematic view on different transfer unit configurations.

Therefore, as a first sub-process, we perform a full scan of the workspace by moving the camera to two different viewpoints ($\alpha_1 = 20°, \alpha_2 = -30°$). During this initial workspace scan, the information of all detected objects is compared with the information available on the virtual representations of TM_1^P and TU^P respectively. The detection method uses a cluster algorithm with the assumption that a cluster of close points belongs to a single object. When objects are placed close together, the clustering algorithm may not be able to distinguish between the objects. This challenge is illustrated in Figure 5.17. In Figure 5.17a, both objects are placed far enough away from each other and can be reliably detected as two separate clusters. In Figure 5.17b, the objects are placed too close to each other and are detected as a single object. The required distance between the objects for successful clustering increases as the camera angle increases, due to more undetected points on the uncaptured back of the objects.



(a) Two objects that can be sufficiently distinguished.

(b) Two objects that are too close together to be distinguished by the clustering algorithm.

Figure 5.17: Multi object detection challenges.

Object Transfer Process

During the design phase of the transfer unit, our aim was to be able to transfer a wide variety of objects without the need for a container or other form of load carriers. In principle, a conveyor belt can achieve this for objects that physically fit onto the belt and create enough friction to cause the object to take on the linear motion of the rotating belt (Pati and Majumdar 2020). When two separate conveyor belts are aligned, such as when the transport module docks

to a transfer unit, the goal is to position the outer pulleys of the belts as close together as possible without creating physical contact. The circular shape of the pulleys creates a gap between the flat surfaces of the conveyor belts that any object must overcome to move from one belt to the other. Figure 5.18 shows the minimum gap area for two perfectly aligned conveyor belts. The size of the gap increases with imperfect alignment, e.g., due to a less accurate docking process. Depending on the shape of the objects, this gap can affect the motion of the object during transfer and, in the worst case, result in an object getting stuck and unable to clear the gap. In the DT application study in Chapter 6, we investigate how the gap affects the transfer of starter motor cores $\mathcal{O}^P$.

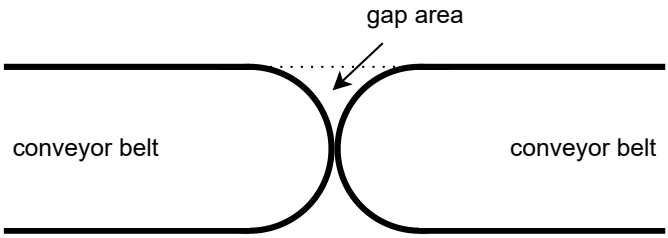


Figure 5.18: Gap area between two aligned conveyor belts, see (Klein and Furmans 2025)

5.6 Chapter conclusion

In this chapter we have described details of an implemented CPS for realising object transfers as part of the *AgiProbot* demonstration system. After a short introduction to the *AgiProbot* project, we presented the implemented physical assets, the transport unit, a vision-enhanced AMR, the transfer units, stationary transfer points at each station, and a set of starter motor cores, which we consider to be the set of objects to be transferred. We further introduced three implemented logical assets, namely the *Transport Module System*, *PropS* and the *Experiment Manager*. For each physical asset, we presented the virtual models that complete the implementation of the CPS based on the architecture described

in Chapter 3. Through interaction, these assets enable the flexible material flow in the *AgiProbot* demonstration system. The transport module handles transport operations that first require the execution of a global navigation and docking process to a target transfer unit. Once docked, the transport module forms an OTS with the transfer unit, which we described based on the formal description introduced in Chapter 4. We further provided implementation details of the detection and tracking method and the visual servoing approach. Finally, we discussed several challenges and limitations of the implemented system. The challenges include inaccuracies in the docking process which result in a gap between the conveyor belts of the docked transport module and the transfer unit which the objects must be able to overcome. In the following chapter, we investigate how these challenges affect the object transfer process and how we can use the virtual models to reduce the imposed uncertainty with respect to successful transfer processes.

6 Digital Twin Application Study

The content of this chapter extends upon sections of a working paper titled "A Study on the Predictive Capabilities of Digital Twins for Object Transfers in Fluid Automated Remanufacturing" (Klein and Furmans 2025). The author of this thesis was responsible for the conceptualisation, methodology, software programming, validation, formal analysis, writing, and visualisation of the research presented in this chapter.

In the previous chapter we presented the implementation of a CPS in which a physical transport module (TM_1^P) navigates and docks to a transfer unit ($TU^P \in \mathcal{TU}^{\mathcal{P}}$) to initiate an OTS to transfer an object, in this case one of the starter motor cores ($o^P \in \mathcal{O}^{\mathcal{P}}$). Due to the challenges outlined in Section 5.5, the object transfer process must be regarded as probabilistic. The success of an object transfer depends on several factors. Based on our preliminary study, the following three factors are the most influential (Klein and Furmans 2023):

- The **docking pose quality**: The quality of the docking process directly defines the shape and size of the gap between the transport module and the transfer unit.
- The **object to be transferred** o^P: Since the objects are transferred without the aid of a container or any other form of load carrier, the interaction between the contour of the object and the gap area between the conveyor belts may result in an object being unable to clear the gap.

- The **pre-transfer object state** s^{pre}: The pre-transfer state of o^P defines which part of the contour enters the gap at which point in time. A carefully chosen pre-transfer state may avoid any issues during the transfer.

In general, it is not the effects of a single of these factor, but the combination of factors that determines whether a transfer is successful or not. The goal of this application study is to use our DTs to predict a single pre-transfer state for each physical o^P that promises to have the highest probability of success. Our DT application study is structured as follows:

1. **Calibration Experiments**: We perform an initial set of physical experiments on a small set of starter motor cores.
2. **Virtual Model Calibration**: We use these results to calibrate our virtual models. In addition, we use the measured docking pose errors to build a synthetic docking pose generator which allows us to sample realistic docking pose errors for further virtual experiments.
3. **Full State Experiments**: We conduct a large series of virtual experiments in which we estimate the probability of success, denoted as $\hat{p}^V$, for each pre-transfer state of each virtual starter motor core o^V.
4. **Determine Best Pre-Transfer State**: Through a series of transformations, we identify the pre-transfer state with the highest probability of success, denoted as $s^{\text{pre}}_{o,\text{best}}$ for each virtual object o^V. We use this state as an estimate for the best state for the corresponding real-world object o^P. To gain additional intuition about the capabilities of our DTs, we similarly determine $s^{\text{pre}}_{o,\text{worst}}$ which is the state with the lowest probability of success.
5. **Validation**: We validate the estimated probabilities of success for $s^{\text{pre}}_{o,\text{best}}$ and $s^{\text{pre}}_{o,\text{worst}}$ by an additional set of real-world experiments.

The chapter is structured as follows. Section 6.1 describes some application study-specific adaptations, including pre-transfer state restrictions and the implementation of docking pose error measurements. Section 6.2 formally describes the real and virtual experiments performed. We discuss the results of

a first set of calibration experiments in Section 6.3 and the proposed method to synthetically generate docking error poses in 6.4. We use the results of the calibration experiments to evaluate and calibrate our virtual models in Section 6.5. In Section 6.6, we conduct a series of virtual full-state experiments using our DTs to estimate the probability of success for each pre-transfer state for the full set of starter motor cores. Finally, we validate and discuss the results by performing an additional set of physical experiments in Section 6.7.

6.1 System Adaptations

In this section we introduce some relevant system adaptations specific to this application study. In Section 6.1.1 we restrict the previously defined set of pre-transfer states to a smaller, discrete set. In Section 6.1.2, we introduce the method for estimating the docking pose error.

6.1.1 Pre-Transfer State Restrictions

In Equation 5.7 we have defined the set of pre-transfer states as follows:

$$\mathcal{S}_o^{\text{pre}} = \left\{ s \in \mathcal{S}_o \;\middle|\; \begin{pmatrix} a = a_{\text{pre}} \\ t = t_0 \\ (x - s_x \geq -0.625m) \wedge (x + s_x \leq 0.0m) \\ (y - s_y \geq -0.2m) \wedge (y + s_y \leq 0.2m) \end{pmatrix} \right\} \tag{6.1}$$

This set contains an infinite number of possible pre-transfer states. For the experiments in this application study, only a limited set of pre-transfer states can be practically evaluated. Therefore, we define the application study specific subset ${}^{E}\mathcal{S}_o^{\text{pre}}$ in the following.

$${}^{E}\mathcal{S}_o^{\text{pre}} \subset \mathcal{S}_o^{\text{pre}} \tag{6.2}$$

First we restrict the position of o, (x, y) to three fixed positions:

$$\begin{aligned} p_{\text{left}} &= (-0.325m, 0.094m), \\ p_{\text{middle}} &= (-0.325m, 0.0m), \\ p_{\text{right}} &= (-0.325m, -0.094m) \end{aligned} \tag{6.3}$$

We have marked the three possible positions in Figure 6.1a. Next, we discretise the object orientation θ to a set of fixed integers that cover a full rotation:

$${}^{E}\Theta = \{\theta \mid \theta \in \mathbb{Z}, -179 \leq \theta \leq 180\} \tag{6.4}$$

If the object is placed in p_{left} or p_{right}, some values in ${}^{E}\Theta$ would cause the object to collide with the area outside the conveyor belts. This is shown in Figure 6.1b. The object is placed with a yaw angle of $\theta = 15°$. An angle of e.g. $\theta = 90°$ in the given position would not be feasible without exceeding the width of the conveyor belt. We therefore split ${}^{E}\Theta$ into two different subsets: ${}^{E}\Theta_{\text{single}}$ for objects states which refer to a single conveyor position (p_{left} and p_{right}) and ${}^{E}\Theta_{\text{both}}$ for objects placed in the centre that can cover the full rotation range.

$${}^{E}\Theta_{\text{single}} = [-179.. - 155] \cup [-25..25] \cup [155..180] \tag{6.5}$$

$${}^{E}\Theta_{\text{both}} = {}^{E}\Theta \setminus {}^{E}\Theta_{\text{single}} \tag{6.6}$$

Next, we define three pre-transfer state subsets, the union of which is ${}^{E}\mathcal{S}_{o}^{\text{pre}}$, the full set of restricted pre-transfer states used in this application study.

$${}^{E}\mathcal{S}_{o}^{\text{pre}} = {}^{E}\mathcal{S}_{o,\text{left}}^{\text{pre}} \cup {}^{E}\mathcal{S}_{o,\text{both}}^{\text{pre}} \cup {}^{E}\mathcal{S}_{o,\text{right}}^{\text{pre}} \tag{6.7}$$

$$
{}^{E}\mathcal{S}^{\text{pre}}_{o,\text{left}} = \left\{ s \in \mathcal{S}^{\text{pre}}_{o} \;\middle|\; \begin{pmatrix} a = a_{\text{pre}} \\ t = t_0 \\ (x,y) = p_{\text{left}} \\ \theta \in {}^{E}\Theta_{\text{single}} \\ c = \text{'left'} \end{pmatrix} \right\} \tag{6.8}
$$

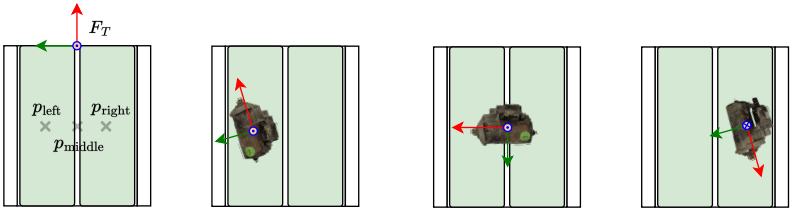
$$
{}^{E}\mathcal{S}^{\text{pre}}_{o,\text{both}} = \left\{ s \in \mathcal{S}^{\text{pre}}_{o} \;\middle|\; \begin{pmatrix} a = a_{\text{pre}} \\ t = t_0 \\ (x,y) = p_{\text{middle}} \\ \theta \in {}^{E}\Theta_{\text{both}} \\ c = \text{'both'} \end{pmatrix} \right\} \tag{6.9}
$$

$$
{}^{E}\mathcal{S}^{\text{pre}}_{o,\text{right}} = \left\{ s \in \mathcal{S}^{\text{pre}}_{o} \;\middle|\; \begin{pmatrix} a = a_{\text{pre}} \\ t = t_0 \\ (x,y) = p_{\text{right}} \\ \theta \in {}^{E}\Theta_{\text{single}} \\ c = \text{'right'} \end{pmatrix} \right\} \tag{6.10}
$$

Combined, the pre-transfer set ${}^{E}\mathcal{S}^{\text{pre}}_{o}$ consists of **924** possible states for each object o. Figure 6.1 shows some examples of included pre-transfer states.

6.1.2 Docking Pose Error Measurement

During the initial calibration experiments, we measure the docking error ${}_{e}\tilde{\xi}$ each time TM^{P}_{1} docks to one of the transfer units, see Section 5.4.1. To measure the docking error, we mount a stationary camera on each of the four transfer units and attach a 5x5 ChArUco board to the transport module. Using a fiducial marker or an array of markers for pose estimation is a widely used method in robotics and VR applications (Kalaitzakis et al. 2021). Figure 6.2 shows the



(a) Pre-transfer positions, see equation 6.3

(b) $s^{\text{pre}} \in {}^{E}\mathcal{S}^{\text{pre}}_{o,\text{left}}$

(c) $s^{\text{pre}} \in {}^{E}\mathcal{S}^{\text{pre}}_{o,\text{both}}$

(d) $s^{\text{pre}} \in {}^{E}\mathcal{S}^{\text{pre}}_{o,\text{right}}$

Figure 6.1: Pre-transfer state examples $s^{\text{pre}} \in {}^{E}\mathcal{S}^{\text{pre}}_{o}$.

installed setup with the overlayed coordinate frames as listed in Table 6.1. To estimate the docking pose error, we need a reference goal pose against which we can compare the currently measured pose. Therefore, we measure an individual reference pose $\boldsymbol{T}_{C_T B_G}$ for each transfer unit after manually docking the transport module using its joystick. After completion of each individual docking process, we measure the current pose of the ChArUco board, $\boldsymbol{T}_{C_T B}$. To make the values easier to interpret, we solve Equation 6.11 and extract the relevant docking error components from $\boldsymbol{T}_{D_G B}$.

$$ {}_e\tilde{\xi} = \begin{pmatrix} {}_e\tilde{\xi}_x \\ {}_e\tilde{\xi}_y \\ {}_e\tilde{\xi}_\theta \end{pmatrix} \leftarrow \boldsymbol{T}_{D_G D} = \boldsymbol{T}_{D_G B_G}\, \boldsymbol{T}^{-1}_{C_T B_G}\, \boldsymbol{T}_{C_T B}\, \boldsymbol{T}^{-1}_{DB} \tag{6.11} $$

Table 6.1: Overview of the relevant coordinate frames used to measure the docking error of TM_1^P.

Frame	Description
F_{C_T}	Fixed frame in the camera module to evaluate the docking pose error, z-axis as optical axis.
F_B	Frame defined at the corner of the detected 5x5 ChArUco board
F_{B_G}	Fixed frame defined at the goal pose of the ChArUco board.
F_D	Fixed frame centered on the transport module.
F_{D_G}	Frame at the goal docking pose.

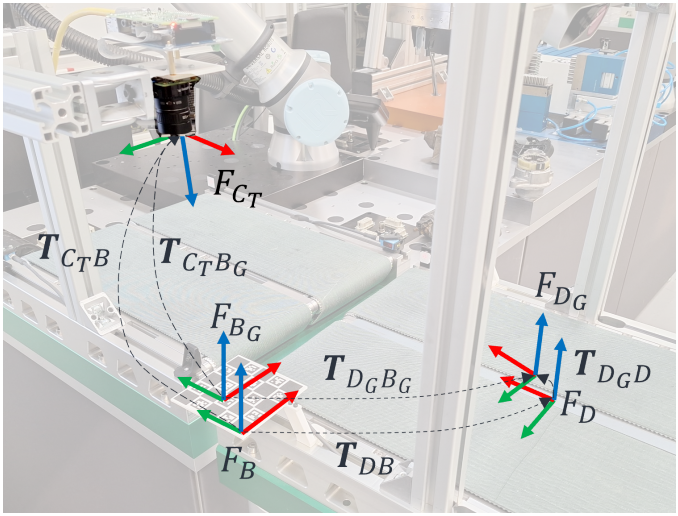


Figure 6.2: Camera-based set up to evaluate the docking pose error.

6.2 Experiment Descriptions

This application study includes several series of real-world and virtual experiments. We describe the standardised process of a single experiment for the real-world case in Section 6.2.1 and for the virtual case in Section 6.2.2.

6.2.1 Real-world Experiment Description

We define a real-world experiment as E^P which consists of the following steps:

1. Global navigation of TM_1^P to the stored coarse navigation pose of the requested target transfer unit $TU^P \in \mathcal{TU}^P$, as described in Section 5.4.1.
2. Execution of the docking procedure and subsequent measurement of the docking pose error ${}_e\tilde{\xi}$, see Section 6.1.2.
3. We manually place the object $o^P \in \mathcal{O}^P$ in a desired pre-transfer state $s^{\text{pre}} \in {}^E\mathcal{S}_o^{\text{pre}}$, see Section 6.1.1.
4. The OTS is being initiated and the object transfer process is executed. First o is transferred from TM_1^P to TU^P:

$$y_1^P = OTS(o^P, TM_1^P, TU^P) \tag{6.12}$$

5. If the transfer was successful ($y_1^P = 1$), the object is transferred back to TM_1^P.

$$y_2^P = OTS(o^P, TU^P, TM_1^P) \tag{6.13}$$

6. Finally, we set the overall result of the experiment:

$$y^P = \begin{cases} 1 & \text{if } y_1^P = 1 \text{ and } y_2^P = 1 \\ 0 & \text{otherwise} \end{cases} \tag{6.14}$$

To summarise, in a real-world experiment E^P, after navigation and docking of the transport module TM_1^P, the manually placed object o^P in a pre-transfer state s^{pre} is first transferred to a transfer unit TU^P, and, if the transfer is successful, returned to TM_1^P. The returned results include the overall transfer result y and the measured docking error:

$$\left(y^P, {}_e\tilde{\xi}\right) = E_i^P\left(o^P, TM_1^P, TU^P, s^{\text{pre}}\right) \tag{6.15}$$

The individual steps of E^P are visualized in Figure 6.3

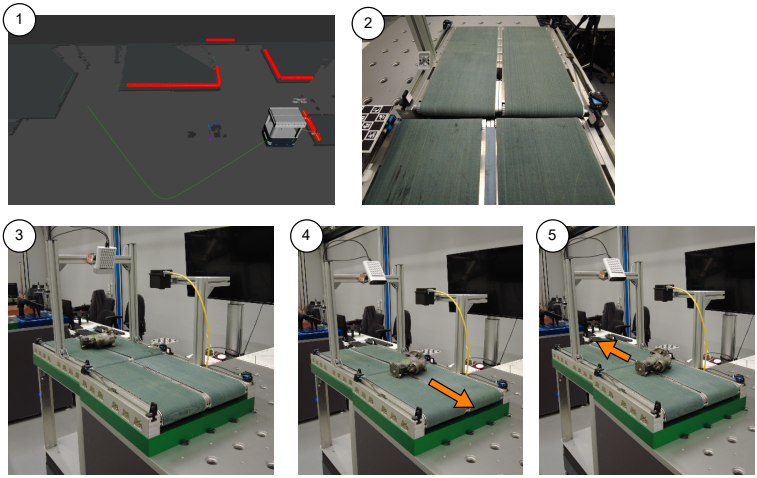


Figure 6.3: Procedure of a real-world object transfer experiment.

6.2.2 Virtual Experiment Description

Compared to the real-world experiment description, the virtual experiment E^V differs in some steps. Since we are not trying to virtually replicate the docking process directly, we immediately set the virtual transport module to a docking pose with with a predefined docking error. A virtual experiment consists of the following steps:

1. Instantiate a virtual transport module TM^V and a virtual transfer unit TU^V.
2. Set the docking pose of TM^V and apply a defined docking pose error ${}_e\xi$.
3. Instantiate a virtual object $o^V \in \mathcal{O}^V$ and set it to the desired pre-transfer state s^{pre}.

4. Perform an object transfer process

$$y_1^V = OTS(o^V, TM^V, TU^V) \tag{6.16}$$

5. If the transfer was successful ($y_1^V = 1$), the object is transferred back to TM^V.

$$y_2^V = OTS(o^V, TU^V, TM^V) \tag{6.17}$$

6. Evaluate overall result, see Equation 6.14.

To summarise, in a virtual experiment E^V, after setting the docking pose and applying a defined docking pose error ${}_e\xi$, the object o^V in state s^{pre} is first transferred to the transfer unit TU^V, and, if the transfer is successful, transferred back to TM^V:

$$y^V = E^V\left(o^V, TM^V, TU^V, s^{\text{pre}}, {}_e\xi\right) \tag{6.18}$$

The individual steps of E^V are visualized in Figure 6.4.

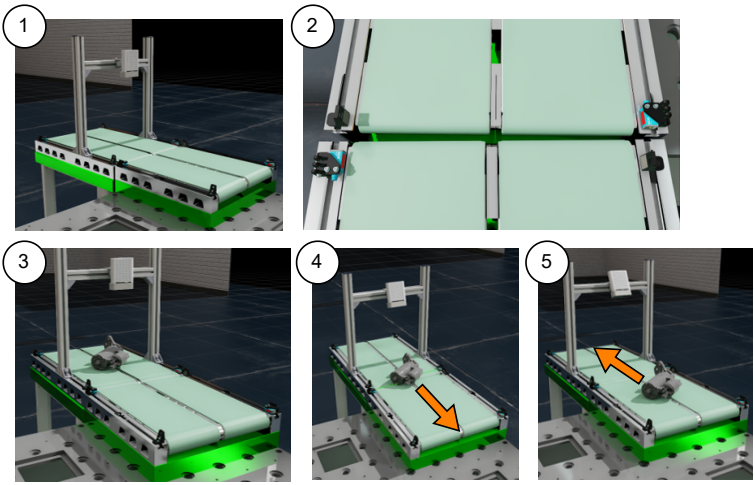


Figure 6.4: Procedure of a virtual object transfer experiment.

6.3 Calibration Experiments

As a first step in this application study, we conduct an initial set of 200 real-world object transfer experiments, see Table 6.2. The results of these experiments are used for two purposes:

- We use the measured docking pose errors to train a synthetic docking error generator to be used in further virtual experiments.
- We use the results y^P to calibrate our virtual models to reduce the existing sim-to-real gap.

For a real-world experiment, we need to define the transfer units to dock to, the objects to transfer and the pre-transfer states of the objects, see Equation 6.15. For the *Calibration Experiments*, we use TM_1^P to dock to all four transfer units in $\mathcal{TU}^P$. We do not consider the full set of objects, but limit the set to four randomly selected objects. The flowchart in Figure 6.5 shows the process used to generate the necessary experimental parameters. Before discussing the quantitative results in Section 6.3.2, we first present and explain the result plots in the following section.

6.3.1 Result Plot Introduction

Figure 6.6 introduces the result plot that is used in the following sections. The centre of the plot shows an image of o. If the engagement relay is on the right side, o has been placed on the face $f =$ 'front'. In general, the binary result y

Table 6.2: Experiment description: *Calibration Experiments*.

Name:	*Calibration Experiments*	**Number of runs:**	1
Type:	physical (E^P)	**Number of experiments per run:**	200
Transport Module(s):	TM_1^P	**Total number of experiments:**	200
Transfer Unit(s):	$TU^P \in \mathcal{TU}^P\}$		
Objects:	$o^P \in \{o_1^P, o_{18}^P, o_{23}^P, o_{32}^P\}$		

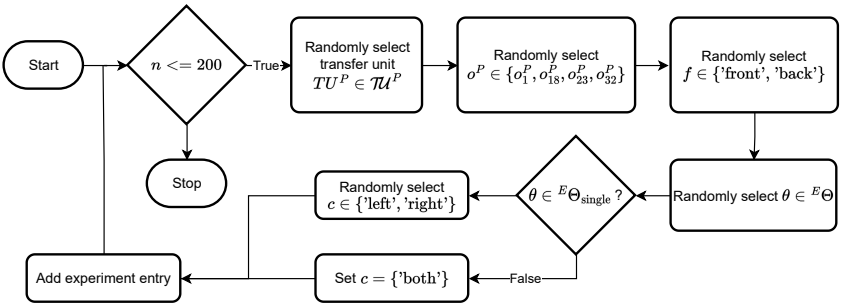


Figure 6.5: Flowchart for generating the *Calibration Experiments*.

of an experiment E is represented by a coloured dot. A green dot indicates a successful transfer experiment ($y = 1$), a red dot indicates an unsuccessful one ($y = 0$). The result dot of an experiment is placed on the figure at a specific position based on the object's pre-transfer state s^{pre}. The figure illustrates three pre-transfer state components, (θ, f, c). Since c implies (x, y), this is sufficient for the restricted set introduced in Section 6.1.1. We apply the angles $\theta \in {}^E\Theta$ to a circular band surrounding the image of o. For the angles $\theta \in {}^E\Theta_{\text{single}}$, we divide the circle into an inner and outer one. In this angle range, two conveyor positions are possible. The outer circle covers results where $c =$ 'left' and the inner circle where $c =$ 'right'.

6.3.2 Quantitative Results and Discussion

Figure 6.7 shows the binary results of the calibration experiments. Table A.1 shows the results in tabular form.

Each visualised data point represents the result of a single experiment E^P in which TM_1^P docked to a specific transfer unit TU^P with a unique docking error, see Equation 6.15. The result of this data point is therefore only certain under the specific conditions of the experiment performed. If TM_1^P had docked in a slightly different way, the result might have been the opposite. Based on the results shown in Figure 6.7, the following general observations can be made:

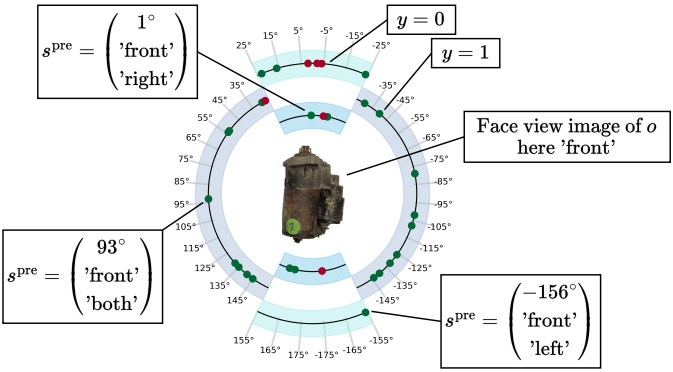
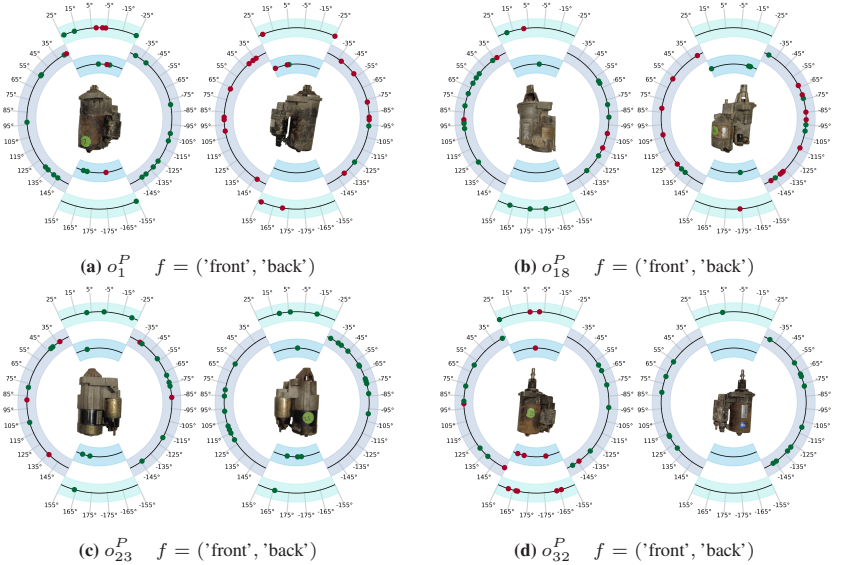


Figure 6.6: Exemplary labeled result plot for o_1^P.

- Each core o^P behaves differently during transfer, presumably due to its unique shape and the interaction of the shape with the gap area between the conveyor belts, as previously mentioned in Section 5.5.
- One can observe some unique patterns for each individual core. For example, the evaluated states of the back face of o_{23}^P and o_{32}^P were all successful. The majority of failed transfers for the front face of o_1^P and o_{33}^P occur for angles $\theta_i \in {}^E\Theta_{\text{single}}$. There is no intuitive pattern that applies to all four motor cores.
- The majority of the failed object transfers are due to the object not being able to pass the gap between the conveyors. Figure 6.8 shows a series of images from four different calibration experiments. In all four depicted cases, the object got stuck and was unable to pass the gap between the belts.

Overall the calibration experiments include (135/200) positive examples and (65/200) negative examples.



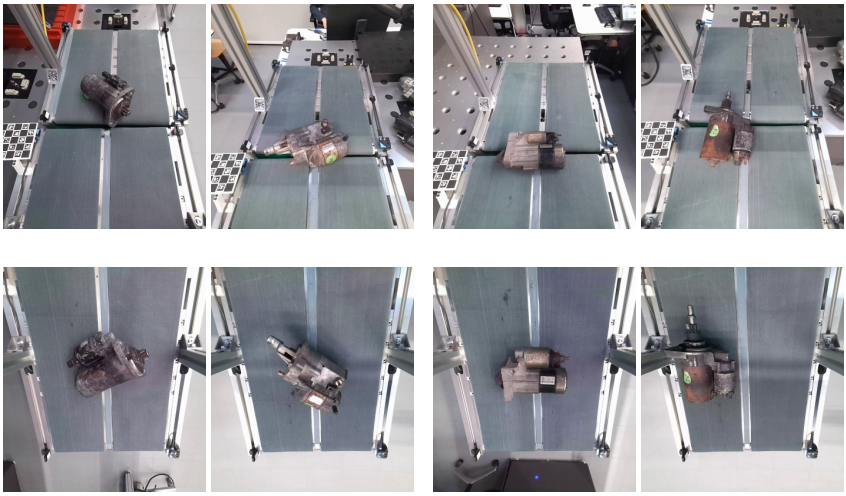
(a) $o_1^P \quad f = (\text{'front'}, \text{'back'})$ **(b)** $o_{18}^P \quad f = (\text{'front'}, \text{'back'})$

(c) $o_{23}^P \quad f = (\text{'front'}, \text{'back'})$ **(d)** $o_{32}^P \quad f = (\text{'front'}, \text{'back'})$

Figure 6.7: Real world experiment results for four different starter motor core variants $\{o_1^P, o_{18}^P, o_{23}^P, o_{32}^P\}$. Each subfigure shows the results for $(f = \text{'front'})$ on the left and $(f = \text{'back'})$ on the right.

Docking Error

For each individual experiment we measured the docking error of TM_1^P, as explained in Section 6.1.2. The docking error plays an important role on the transfer results, since it influences the size of the gap between the docked conveyor belts. Figure 6.9 shows the measurement results. The top row shows the accumulated histograms for all three measured docking error components $({}_e\tilde{\xi}_x, {}_e\tilde{\xi}_y, {}_e\tilde{\xi}_\theta)$. The histograms of all three components are approximately bell-shaped. For ${}_e\tilde{\xi}_x$ and ${}_e\tilde{\xi}_y$, the values are in an expected range. The histogram of ${}_e\tilde{\xi}_\theta$ is shifted towards positive yaw angle errors. This implies that the gap between the conveyors belts is not equally wide and that objects in a pre-transfer state with $c = \text{'left'}$ have to overcome a larger gap compared to objects with $c = \text{'right'}$. The four rows below show the measured errors for each individual transfer unit. Here we can observe the difference in the docking quality based on



(a) o_1^P $s^{\text{pre}} = \begin{pmatrix} -52^\circ \\ \text{'back'} \\ \text{'both'} \end{pmatrix}$ **(b)** o_{18}^P $s^{\text{pre}} = \begin{pmatrix} 57^\circ \\ \text{'back'} \\ \text{'both'} \end{pmatrix}$ **(c)** o_{23}^P $s^{\text{pre}} = \begin{pmatrix} 88^\circ \\ \text{'front'} \\ \text{'both'} \end{pmatrix}$ **(d)** o_{32}^P $s^{\text{pre}} = \begin{pmatrix} -2^\circ \\ \text{'front'} \\ \text{'left'} \end{pmatrix}$

Figure 6.8: Images series of selected *Calibration Experiment* result states: The lower row shows the pre-transfer states at t_0, the upper row shows the final transfer state at t_{end}. For all selected experiments, the transfer failed: $s(t_{end}) \in \mathcal{S}_o^-$. The binary result is also shown in Figure 6.7

.

the station module configuration, as discussed previously in Section 5.5. While ${}_e\tilde{\xi}_\theta$ is similarly distributed across all transfer units, there is a clear difference in ${}_e\tilde{\xi}_y$ between TU_A^P and TU_D^P compared to TU_B^P and TU_C^P.

The three docking pose error components are not independent. This can be observed in the 3x3 correlation matrix shown in Figure 6.10 with the largest correlation between the x-translation and the y-translation.

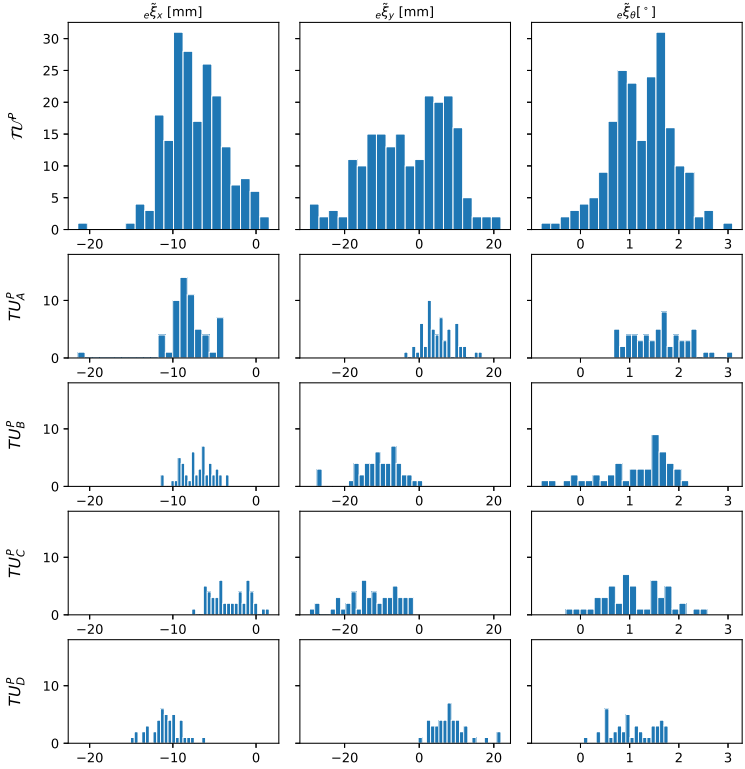


Figure 6.9: Histograms of the measured docking errors.

Figure 6.10: Correlation matrix of the measured docking errors.

6.4 Synthetic Docking Error Generator

In the real-world object transfer system, the docking process is not perfectly accurate. TM_1^P stops in front of a transfer unit with a certain docking error ${}_e\tilde{\xi}$, whose measurements we evaluated in the previous section. Since we are more interested in the transfer process itself, we do not simulate the docking process virtually but abstract it by directly setting the final docking pose of a virtual transport module TM^V, see step 2 in the virtual experiment description in Section 6.2.2. Therefore, we want to be able to synthetically generate new docking errors from a distribution similar to the distribution of the real docking error. Since the docking pose error components $({}_e\xi_x, {}_e\xi_y, {}_e\xi_\theta)$ are not independent, the synthesiser needs to capture the joint distribution.

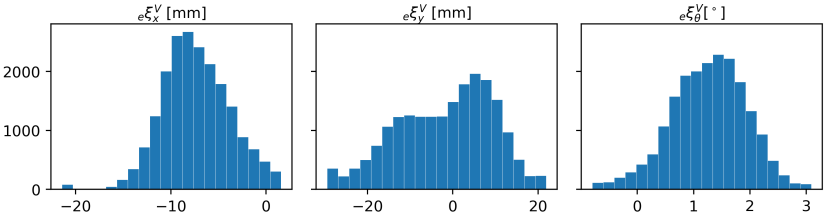
To achieve this, we use the *Synthetic Data Vault*, an open source framework for training generative models based on tabular data (Patki et al. 2016). The approach constructs a Gaussian copula that is able to describe the dependence structure between random variables while preserving their marginal distributions. The Gaussian copula for our case of three random variables is defined as follows (Kächele 2023): Let Φ_ρ be the *Cumulative Distribution Function* (CDF) of a multivariate normal distribution with standard normal margins and a 3×3 correlation matrix $\boldsymbol{\rho}$, and ϕ^{-1} the inverse of the univariate standard normal CDF. The Gaussian copula is then given by

$$C(u_1, u_2, u_3) = \Phi_\rho\{\phi^{-1}(u_1), \phi^{-1}(u_2), \phi^{-1}(u_3)\} \quad \text{with} \quad u_i = F_i(X_i) \tag{6.19}$$

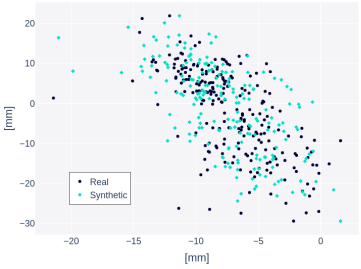
In a first step, the approach tries to fit a set of distribution functions (Gaussian, truncated Gaussian, Beta, Student t, etc.) to obtain an estimated marginal distribution for each random variable $F_i(X_i)$. These marginal distributions are then transformed using the inverse CDF of the standard normal distribution ϕ^{-1}, resulting in three standard normal distributed variables. These are then combined into a multivariate normal distribution in which the correlation matrix $\boldsymbol{\rho}$ describes the dependence structure between the variables. By inverting the process, samples from the multivariate distribution can be transformed back into

their original form. For a detailed description of copulas, see (Durante and Sempi 2016).

Figure 6.10 shows the results for synthetically generated samples. Figure 6.11a shows histograms for each marginal distribution. The shapes are similar to the measured ones, see Figure 6.9. We can see that the correlation between the variables has also been accurately preserved , see Figure 6.11b and Figure 6.11a.



(a) Histogram of 20000 synthetic generated docking pose errors



(b) Correlation matrix between the individual docking pose error components

(c) Correlation matrix between the individual docking pose error components

Figure 6.11: Synthetically generated docking pose errors.

Synthetically generated docking errors are used in the virtual *Full State* experiments in Section 6.6.

6.5 Virtual Model Tuning and Evaluation

To enable the forecasting function of our DTs, we use our virtual models introduced in Section 5.2.1 to simulate object transfer processes. The transferability of the results of the virtual object transfers to the real-world is highly dependent on the quality of the virtual models. Our goal in this section is to find suitable parameters for the virtual models to mimic the real-world object transfer sufficiently to allow us to use the results in the decision-making process of our real-world system.

Table 6.3 summarises the experiments that we refer to as *Virtual Calibration Experiments*. In a single *Virtual Calibration Experiment* E^V (see Equation 6.18) we replicate the *Calibration Experiments* E^P:

- We instantiate a virtual transport model TM^V and transfer unit TU^V that form an OTS to transfer a virtual starter motor o^V placed with the pre-transfer state s^{pre} used in the *Calibration Experiments*.
- We use the measured docking error ${}_e\tilde{\xi}$ of the corresponding real-world experiment E^P and apply it to E^V.

We define a *Virtual Calibration Run* as a set of experiments $\{E_1^V, E_2^V, ..., E_{200}^V\}$ with a fixed set of parameters. Due to the inherently non-deterministic behaviour of physics engines, we cannot rely on the results of a single *Virtual Calibration Run*. This variability can be due to several factors, such as the internal handling of floating-point arithmetic, the order of operations within the engine, or the use

Table 6.3: Experiment description: *Virtual Calibration Experiments*.

Name:	*Virtual Calibration Experiments*	**Number of parameter combinations:**	79
Type:	virtual (E_i^V)	**Number of runs:**	30
Transport Module(s):	$TM_i^V \in \mathcal{TM}^V$	**Number of experiments per run:**	200
Transfer Unit(s):	$TU_i^V \in \mathcal{TU}^V$	**Total number of experiments:**	474.000
Objects:	$o^V \in \{o_1^V, o_{18}^V, o_{23}^V, o_{32}^V\}$		

of random number generators for certain calculations. These factors introduce small variations into the simulation, leading to slightly different outcomes each time the experiment is conducted. This is also the case with PhysX, the engine used by NVIDIA Isaac Sim, which we use as the environment in which our virtual models exist (NVIDIA Cooperation 2024). During initial virtual experiments, we found that by starting the simulation with the same input and seeded random number generators, we could generate deterministic results. However, running the same simulation on a different machine or changing the order of the experiments can lead to slightly different behaviour, which subsequently can lead to different experimental results. To obtain reliable results it is essential to run a *Virtual Calibration Run* several times and analyse the distribution of the results.

We treat the comparison between a *Virtual Calibration Run* and the corresponding real-world experiments as a binary classification problem. Due to class imbalances in the calibration experiment results, using accuracy as the main metric can be misleading. Therefore, we use the $F_{0.5}$ score as our main metric to evaluate the fit. The $F_{0.5}$ score is a weighted harmonic mean between precision and recall, with precision weighted twice as much as recall. By choosing this metric, we want to ensure that the virtual models are tuned more conservatively, as more emphasis on precision reduces the number of false positives. Since the virtual models should act as a predictor for the real-world system, we would rather underestimate the outcome of an object transfer than incorrectly predict a false positive. Appendix A.2 derives the $F_{0.5}$ score based on the confusion matrix.

We divide the general tuning process in two subsequent steps. We first examine a general set of physics-engine specific parameters and their influence on the virtual object transfer results in Section 6.5.1. We then tune two parameters specific to our case in Section 6.5.2. We do a final comparison of the virtual object transfer results and conclude the tuning process in Section 6.5.3.

6.5.1 Simulation Parameter Tuning

We initially selected 16 different simulation parameters that may have an influence on the virtual transfer results. All of the parameters are specific to the physics engine PhysX. During a pre-study, 7 of the 16 parameters were found to have no direct influence on the object transfer results. We listed these parameters and their default values which are used in the following in Table A.3. Since we do not explain each individual parameter here, we provide hyperlinks in the referenced tables which link to the engine's documentation.The PhysX version used in this application study is 5.3.1. For the remaining 9 parameters, we conducted sets of controlled experiments, in which we only varied a single parameter under consideration. In total, we evaluated 20 parameter combinations. We assigned a key (ID-S) to each parameter set. Table A.4 shows this set. Due to the non-determinsitic behaviour, each *Virtual Calibration Run* for a given parameter combination was repeated 30 times. Figure 6.12 and Table A.6 show the results. With regards to the *Physics Step Frequency*, lead, for our case to higher $F_{0.5}$ score means with the cost of increased computational efforts. With the regards to the *Collision System*, PCM leads to a higher score. For the *Solver Type*, TGS is prefarable. *GPU Dynamics* enabled has a negative impact on the $F_{0.5}$ score. A negative impact can also be observed for a disabled *Stabilization*. For the *Broadphase Type*, *Min Velocity Iteration Count*, *Friction Offset Threshold*, and *Friction Type*, no significant differences in the parameter values could be observed. Based on the results, we decided to further use parameter set ID-S 2, which features all benefitially evaluated parameters.

6.5.2 Instance Parameter Tuning

While the parameters evaluated in the previous section were rather general with respect to PhysX, in this section we evaluate two parameters that are particularly important for the object transfer system under consideration. We will therefore call them instance parameters. In our pre-study we analysed the behaviour of a small set of wooden objects with a simple shape during object transfer (Klein

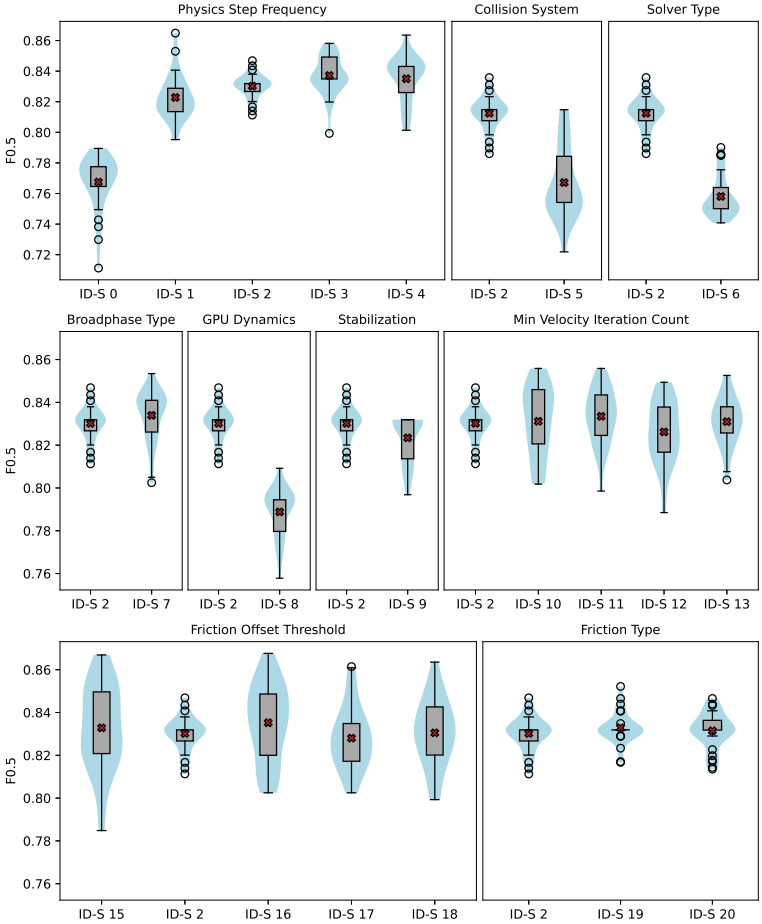


Figure 6.12: Results of *Virtual Calibration Experiments* for different simulation parameters, see Table A.6. Each parameter set (ID-S X) has been simulated 30 times. The red cross shows the estimated mean $F_{0.5}$ value. The box extends from the first quartile (Q1) to the third quartile (Q3) of the data. The whiskers extend from the box to the furthest data point within 1.5x the interquartile range (IQR) of the box. Flier points are those beyond the end of the whiskers. The overlayed violin plot shows the result of a kernel density estimation which estimates the probability density function of the score value.

and Furmans 2023). The study was carried out on the same conveyor belts that are used on the transport module and the transfer unit in this application study. We found that two parameters had a significant effect on the transferability of the objects:

- **Dynamic Friction Coefficient** (μ_k): Friction is the main reason why objects move on a conveyor belt. When the objects need to overcome the conveyor belt gap, the amount of friction force generated might decide if the object is able to overcome the gap or not.
- **Initial Gap Distance** (d_{gap}): In Section 6.1.2, we explained that the docking pose error is measured relative to a manually set goal pose (F_{B_G}). d_{gap} is used to initially set the virtual conveyor belts apart before applying a docking pose error. Figure 6.13 illustrates the parameter.

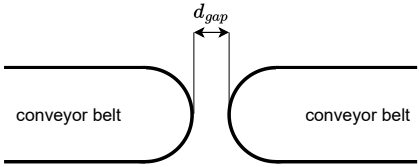
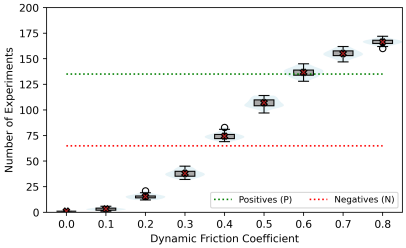


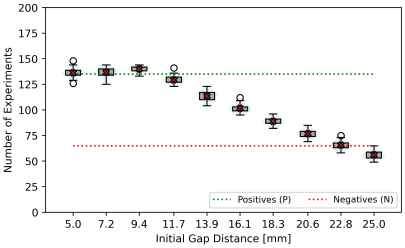
Figure 6.13: Visualisation of the *Inital Gap Distance* parameter.

To get an intuition of the influence of both parameters on the object transfer results, we performed a series of virtual experiments in which we varied each parameter individually. Figure 6.14a shows the results for μ_k. As expected, at a value of 0.0, no object can be transferred as the object is not sufficiently set in motion due to the limited friction. By increasing μ_k, more and more object transfers are successful, with the number of positive experiments exceeding the real world results at a value of around 0.6. Figure 6.14b shows the results for a varied d_{gap}. As expected, an increased gap distance leads to fewer successful object transfers.

To find a suitable set of values, we used a Latin hypercube design to generate parameter tuples with a sample size of 20 data points, see table A.5. Figure 6.15a



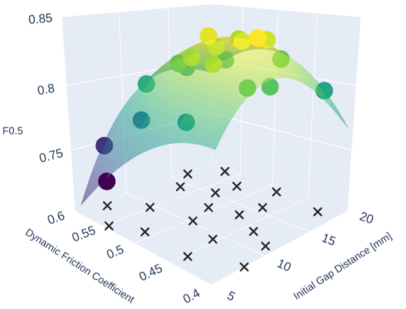
(a) Virtual experiment results for a varied *Dynamic Friction Coefficient*.



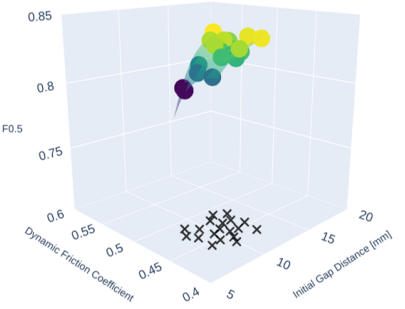
(b) Virtual experiment results for a varied *Initial Gap Distance*.

Figure 6.14: Experiment results for single instance parameter variations

visualises the results. Given the results, we repeated the procedure by generating a second design based on a limited parameter range for both parameters. Table A.7 presents the results in tabular form including standard deviations for all computed metrics. Based on the results, we find the parameter set (ID-I 23) as the set with the highest mean $F_{0.5}$ score. The tuple (ID-S 2, ID-I 23) forms the combined selected parameter set that is used for all following virtual experiments.



(a) Latin hypercube based experiments for both instance parameters.



(b) Second experiment series with a limited parameter design.

Figure 6.15: Results for different instance parameter combinations. The blobs represent the mean F.05 score for the given set of parameters.

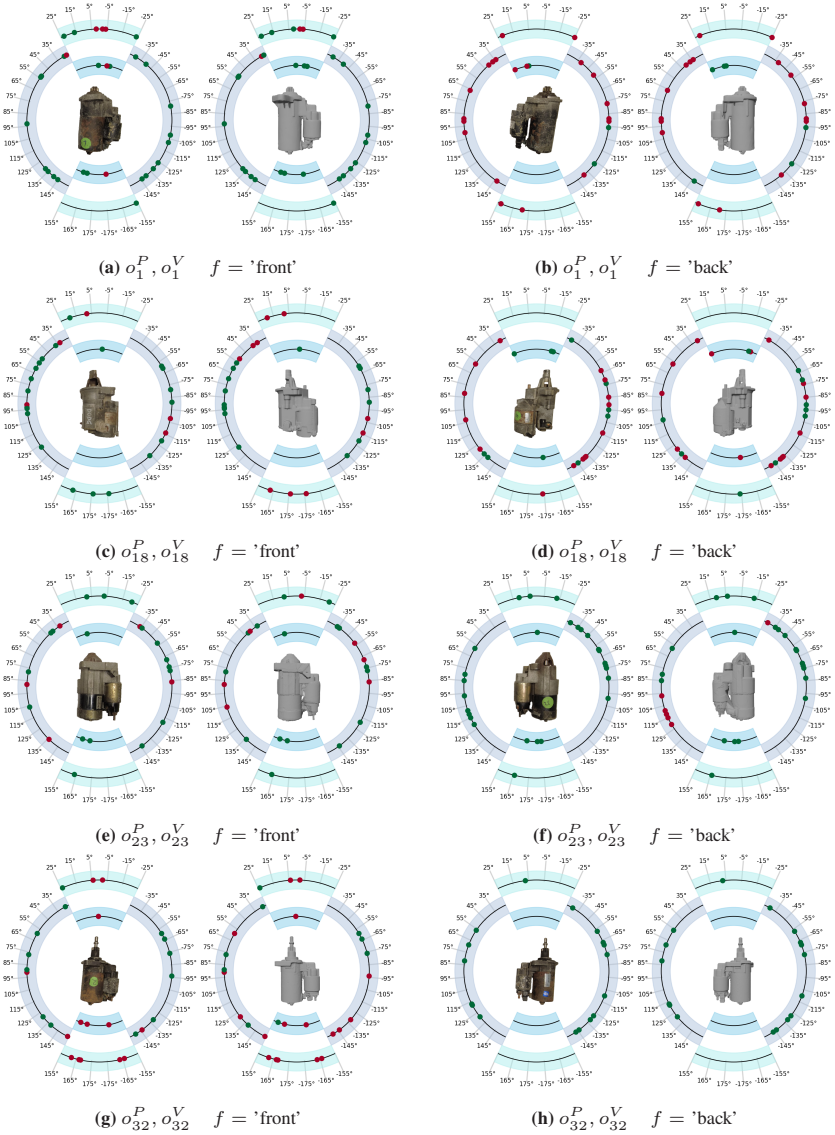
6.5.3 Sim-to-Real Comparison

Table 6.4 shows the numerical results for the selected parameter set (ID-S 2, ID-I 23). We obtain a mean score of $F_{0.5} \approx 0.84$ with a higher precision than recall, indicating a more conservative model in predicting positive results. This is underlined by the number of *False Positives* (FP) being lower then the *False Negatives* (FN). The accuracy is still reasonably high (≈ 0.76), suggesting that the virtual models are performing well overall in simulating object transfers.

Table 6.4: Metrics of a single series of *Virtual Calibration Runs* with the selected parameter sets (ID-S 2, ID-I 23).

PP	PN	FP	FN	TP	TN	accuracy	precision	recall	$F_{0.5}$
123.5	**76.5**	**18.0**	**29.5**	**105.5**	**47.0**	**0.7627**	**0.8548**	**0.7815**	**0.8389**
± 4.3	± 4.3	± 2.8	± 3.1	± 3.1	± 2.8	± 0.021	± 0.019	± 0.023	± 0.017

Figure 6.16 shows the comparison of a single *Virtual Calibration Run* with the *Calibration Experiment* results. Besides some inaccuracies, we can observe that the transfer results of the virtual models are similar to the results of the real-world experiments. Some greater differences can be observed with a number of neighbouring inaccuracies, e.g. for $f =$ 'front' of o_{18}, see Figure 6.16c. The *Calibration Experiments* include three experiments for $c =$ 'left' with $\theta = \{163°, 176°, -174°\}$. All three transfers were successful during the real-world experiments but failed virtually. Similarly, for $f =$ 'back' of o_{23}, see Figure 6.16f, we can observe a set of inaccuracies in the angle range $\theta = \{90°, ..., 120°\}$.



(a) $o_1^P, o_1^V \quad f =$ 'front' **(b)** $o_1^P, o_1^V \quad f =$ 'back'

(c) $o_{18}^P, o_{18}^V \quad f =$ 'front' **(d)** $o_{18}^P, o_{18}^V \quad f =$ 'back'

(e) $o_{23}^P, o_{23}^V \quad f =$ 'front' **(f)** $o_{23}^P, o_{23}^V \quad f =$ 'back'

(g) $o_{32}^P, o_{32}^V \quad f =$ 'front' **(h)** $o_{32}^P, o_{32}^V \quad f =$ 'back'

Figure 6.16: Sim-To-Real result comparison for a single virtual calibration run with parameter set (ID-S 2, ID-I 23). Each row shows the real and the simulated results for $f =$ 'front' (left) and $f =$ 'back' (right).

6.6 Virtual Full-State Experiments

In this section, we use our virtual models to simulate the object transfer of the object set $\mathcal{O}^V$. For each object we simulate the transfer for each pre-transfer states ${}^E s^{\text{pre}} \in {}^E\mathcal{S}_o^{\text{pre}}$. Table 6.5 summarises the set of experiments which we refer to as the *Full State Experiments*. For each experiment, we use a synthetically generated docking pose as described in Section 6.4. We repeat each *Full State Run* 50 times. Therefore, this section contains the results of 1.709.400 virtual object transfer experiments.

$$|E^V| = |\mathcal{O}^V| \cdot |\,{}^E\mathcal{S}_o^{\text{pre}}| \cdot 50 = 37 \cdot 924 \cdot 50 = 1.709.400 \tag{6.20}$$

We first introduce four different pre-transfer state subsets:

$${}^E\mathcal{S}_{o,\text{front,left}}^{\text{pre}} = \left\{ s \in {}^E\mathcal{S}_o^{\text{pre}} \;\middle|\; \begin{pmatrix} f = \text{'front'} \\ c = \{\text{'both'}, \text{'left'}\} \end{pmatrix} \right\} \tag{6.21}$$

$${}^E\mathcal{S}_{o,\text{front,right}}^{\text{pre}} = \left\{ s \in {}^E\mathcal{S}_o^{\text{pre}} \;\middle|\; \begin{pmatrix} f = \text{'front'} \\ c = \{\text{'both'}, \text{'right'}\} \end{pmatrix} \right\} \tag{6.22}$$

$${}^E\mathcal{S}_{o,\text{back,left}}^{\text{pre}} = \left\{ s \in {}^E\mathcal{S}_o^{\text{pre}} \;\middle|\; \begin{pmatrix} f = \text{'back'} \\ c = \{\text{'both'}, \text{'left'}\} \end{pmatrix} \right\} \tag{6.23}$$

Table 6.5: Experiment description: *Full State Experiments*.

Name:	*Full State Experiments*	**Number of runs:**	50
Type:	virtual (E^V)	**Number of experiments per run:**	924
Transport Module(s):	$TM^V \in \mathcal{TM}^V$	**Total number of experiments:**	1.709.400
Transfer Unit(s):	$TU^V \in \mathcal{TU}^V$		
Objects:	Full virtual set $\mathcal{O}^V$		

$$ {}^{E}\mathcal{S}^{\text{pre}}_{o,\text{back,right}} = \left\{ s \in {}^{E}\mathcal{S}^{\text{pre}}_{o} \;\middle|\; \begin{pmatrix} f = \text{'back'} \\ c = \{\text{'both'}, \text{'right'}\} \end{pmatrix} \right\} \tag{6.24} $$

Each of the subsets comprises exactly 360 states and covers a full rotation of the object ($\theta \in {}^{E}\Theta$). The subsets differ with respect to the face on which the object is placed and the single conveyor which is being used for $\theta \in {}^{E}\Theta_{\text{single}}$. For each of the introduced subsets and each run r, we can describe the results for the virtual motor core o^V as series of pairs (y^V, θ):

$$ Y_{o^V,r,\text{front,left}} = \left\{ (y^V, \theta) \;\middle|\; \begin{matrix} y^V \in \{0,1\} \\ \theta \in {}^{E}\Theta \end{matrix} \right\} \tag{6.25} $$

Figure 6.17 shows the plotted results for $Y_{o_1^V,1,\text{front,left}}$.

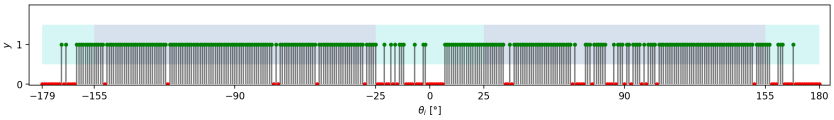


Figure 6.17: Binary results for a single *Full State Run*: $Y_{o_1^V,1,\text{front,left}}$.

Next, we can calculate the mean values of y^V over all 50 runs:

$$ \overline{Y}_{o^V,\text{front,left}} = \left\{ (\overline{y}^V, \theta) \;\middle|\; \begin{matrix} \overline{y}^V = \frac{1}{50} \sum_{r=1}^{50} y_r^V \\ \theta \in {}^{E}\Theta \end{matrix} \right\} \tag{6.26} $$

Figure 6.18 shows the mean results for o_1^V. The plot shows values of $\overline{y}^V \in [0,1]$ which we can interpret as the estimated probability of an object transfer success for the given pre-transfer state.

To select $s^{\text{pre}}_{o,\text{best}}$, our goal is generally to find a pre-transfer state with a high value of $\overline{y}^V$. At the same time we want to avoid choosing a state with a high value of $\overline{y}^V$ which is surrounded by states with low values of $\overline{y}^V$. There are two practical reasons for this:

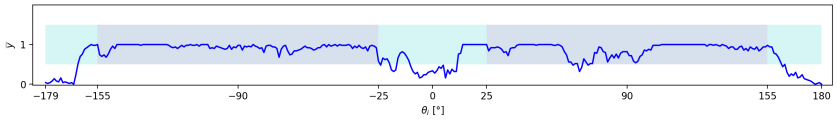


Figure 6.18: Mean transfer results $\overline{Y}_{o_1^V,\text{front,left}}$.

- In the real-world case, a handling robot, which is controlled to place o^P in a proposed pre-transfer state may not achieve the required accuracy.
- Since we know that there is a sim-to-real gap, a wider range of neighbouring states (in terms of θ) with high values for $\overline{y}^V$ suggests that the real probability for the given state is indeed high.

To inherently cope with this reason we smooth the data series by applying a centred moving average filter with a window size of 21:

$$\overline{Y}^{MA}_{o^V,\text{front,left}} = \left\{ \left(\overline{y}_i^{MA}, \theta_i\right) \;\middle|\; \begin{array}{c} \overline{y}_i^{MA} = \frac{1}{21}\sum_{j=-10}^{10} \overline{y}_{i+j} \\ \theta_i \in {}^E\Theta \\ i = 1, 2, \ldots, 360 \end{array} \right\} \tag{6.27}$$

Figure 6.19 shows the resulting plot after applying the filter. A value of $\overline{y}^{MA} = 1$ means that all pre-transfer state in an angle range $[\theta - 10°, \theta + 10°]$ have a value of $\overline{y}^V = 1$. Figure 6.19 shows the smoothed curve. In the above example we

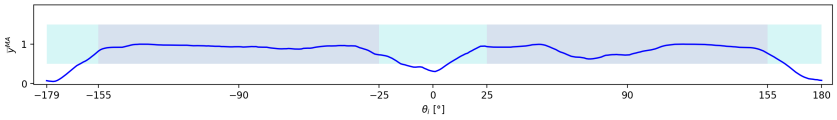


Figure 6.19: Results with applied moving average filter $\overline{Y}^{MA}_{o_1^V,\text{front,left}}$.

were only considering the results for pre-transfer states $s \in {}^E\mathcal{S}^{\text{pre}}_{o,\text{front,left}}$. We

however apply the same procedures to three other subsets as well. For each object o^V we have a set of four data series:

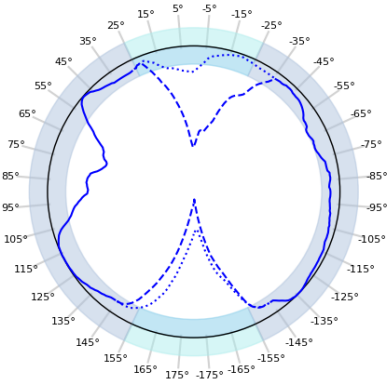
$$Y_{o^V} = \{\overline{Y}^{MA}_{o^V,\text{front,left}}, \overline{Y}^{MA}_{o^V,\text{front,right}}, \overline{Y}^{MA}_{o^V,\text{back,left}}, \overline{Y}^{MA}_{o^V,\text{back,right}}\} \tag{6.28}$$

We can visualise the full result set similar to the plots introduced in Section 6.3.1 by transforming the flat plots to a circular space. Figure 6.20 shows the combined plots for the resulting data series. Each subplot shows two of the four result series. For angles $\theta \in {}^{E}\Theta_{\text{both}}$, both result plots are similar and overlaying. For angles $\theta \in {}^{E}\Theta_{\text{single}}$, the plot divides. The thick dashed plot refers to $\overline{Y}^{MA}_{o^V,\text{front,left}}$ and $\overline{Y}^{MA}_{o^V,\text{back,left}}$ respectively. The thin dotted plot refers to $\overline{Y}^{MA}_{o^V,\text{front,right}}$ and $\overline{Y}^{MA}_{o^V,\text{back,right}}$ respectively.

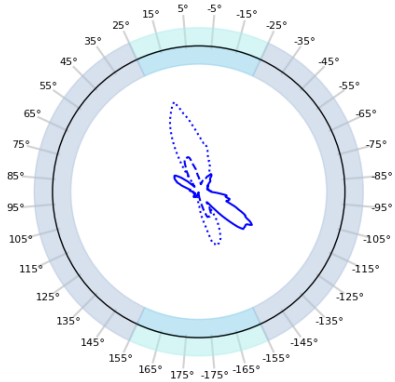
The plots allows a quick intuition on the overall results. In general, when the plot touches the outer circle, then $\overline{y}^{MA} = 1$ for the respective value of θ. As closer the plot is to the center, the smaller the value of $\overline{y}^{MA}$. For the front side (Figure 6.20a), we can e.g. observe promising pre-transfer states in the angle ranges $[110°, 125°]$ or $[-137°, -128°]$. We can also observe that all states of o^V_1 placed on the back side are not promising due to their low values. In the following section we introduce our method to use the results to determine the most promising pre-transfer state that we would us as a proposed state for the real-world system.

6.6.1 Determining Best Pre-Transfer State

As introduced in Section 2.2, a DT is primarily used as a decision support tool. In our application case we aim to use our DTs to suggest, for a given object o, a single pre-transfer that has the most promising probability of success. We define this state as $s^{\text{pre}}_{o,\text{best}}$. Note that this state is only arguably the "best" state according to the procedure that we use to determine it. To find $s^{\text{pre}}_{o,\text{best}}$, we use a two-step approach:



(a) $\{\overline{Y}^{MA}_{o^V_1,\text{front,left}}, \overline{Y}^{MA}_{o^V_1,\text{front,right}}\}$

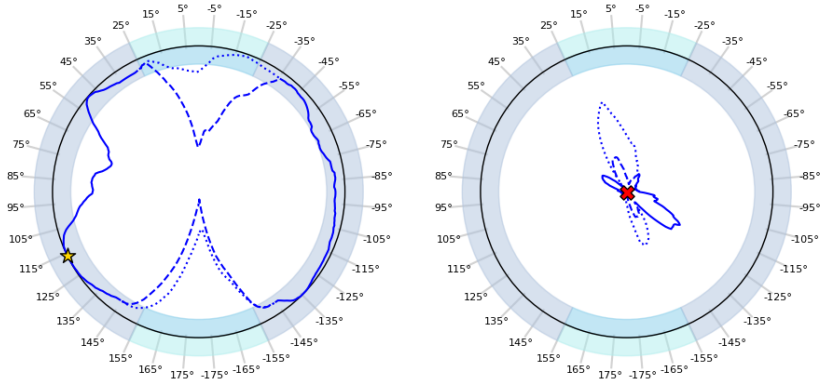


(b) $\{\overline{Y}^{MA}_{o^V_1,\text{back,left}}, \overline{Y}^{MA}_{o^V_1,\text{back,right}}\}$

Figure 6.20: Exemplary result plot for the *Full State Experiments* of o^V. The left plot shows the results for pre-transfer states with $f =$ 'front', the right side for $f =$ 'back'.

- We first determine the best local state for each data series $\overline{Y}^{MA}_{o^V} \in Y_{o^V}$, see algorithm 1.
- We use the local results and compare them to find the global best state, see algorithm 2.

Given a single result series $\overline{Y}^{MA}_{o^V} \in Y_{o^V}$, we first filter for all states that share the maximum value of $\overline{y}^{MA}$. Taking the example plot for o^V_1 in Figure 6.21a, several states were estimated with $\overline{y}^{MA} = 1$. If only a single state has the maximum value, we would be done and have found the best local state. For this example case we enter the nested while loop. We take each of the states with the maximum value and perform a moving average filter similar to Equation 6.27. In essence, we are comparing the value of neighbouring states to evaluate their combined values of $\overline{y}^{MA}$. We only keep state entries if they still have equally high values. With each repetition of this procedure we increase the window size, so that more and more neighbouring states are taken into account. We end up with a single state $s^{\text{pre}}_{o,\text{best}}$ which not only has the highest value for $\overline{y}^{MA}_i$ but also the most neighbouring states with equally high values.



(a) The yellow star indicates $s^{\text{pre}}_{o,\text{best}}$

(b) The red cross indicates $s^{\text{pre}}_{o,\text{worst}}$

Figure 6.21: Examples for $s^{\text{pre}}_{o,\text{best}}$ and $s^{\text{pre}}_{o,\text{worst}}$ determined for o^V_1.

We repeat this procedure for each data series $\overline{Y}^{MA}_{o^V} \in Y_{o^V}$ resulting in four local candidates for $s^{\text{pre}}_{o,\text{best}}$. In algorithm 2, we compare these four candidates and select the one with the highest value of $\overline{y}^{MA}$ and, if multiple states share this value, we select the one with the highest value of the applied window size during the calculation. The star in Figure 6.21a indicates the determined best global state $s^{\text{pre}}_{o,\text{best}}$.

A similar procedure can be performed to determine $s^{\text{pre}}_{o,\text{worst}}$. We provide the algorithm to determine $s^{\text{pre}}_{o,\text{worst}}$ in the appendix, see algorithm 3 and 4.

Figure A.1 shows the result plots for all objects $o^V \in \mathcal{O}^V$. The plots show unique results for each of the objects which demonstrates the complexity and difficulty in predicting the behaviour of each individual starter motor core. We validate the *Full State Experiment* results in the following section.

Algorithm 1 Find Best Pre-Transfer State (Local)

Input: Single result set $\overline{Y}^{MA} \in Y_{o^V}$, see Equation 6.28.
Output: BestStateLocal, AppliedHalfWindowSize

```
MaxValue ← −∞
MaxEntries ← ∅
for each (ȳ^MA, θ) ∈ Ȳ^MA do
    if ȳ^MA > MaxValue then
        MaxValue ← ȳ^MA
        MaxEntries ← {(ȳ^MA, θ)}
    else if ȳ_i^MA = MaxValue then
        Add (ȳ_i^MA, θ_i) to MaxEntries
    end if
end for
HalfWindowSize ← 1
while |MaxEntries| > 1 do
    MaxFilteredValue ← −∞
    FilteredEntries ← ∅
    for each (ȳ^MA, θ) ∈ MaxEntries do
        Apply moving average filter on initial set Ȳ^MA
        FilteredValue ← 1/(HalfWindowSize∗2+1) Σ_{j=−HalfWindowSize}^{HalfWindowSize} ȳ_{i+j}^MA
        if FilteredValue > MaxFilteredValue then
            MaxFilteredValue ← FilteredValue
            FilteredEntries ← {(ȳ^MA, θ)}
        else if FilteredValue = MaxFilteredValue then
            Add (ȳ_i^MA, θ_i) to FilteredEntries
        end if
    end for
    MaxEntries ← FilteredEntries
    HalfWindowSize ← HalfWindowSize + 1
end while
assert |MaxEntries| = 1
BestStateLocal ← MaxEntries[1]
AppliedHalfWindowSize ← HalfWindowSize[1]
```

Algorithm 2 Find Best Pre-Transfer State (Global)

Input: Full result set Y_{o^V}, see Equation 6.28.
Output: BestStateGlobal

```
BestStateGlobal ← (−1, −1) ▷ Initialize BestStateGlobal with a placeholder
MaxValueGlobal ← −∞
MaxHalfWindowSize ← 0
for all Ȳ^MA ∈ Y_oV do
    LocalResult ← Call Algorithm 1 with Ȳ^MA as input .
    BestStateLocal ← LocalResult[1]
    AppliedHalfWindowSize ← LocalResult[2]
    if BestStateLocal[1] > MaxValueGlobal then
        BestStateGlobal ← BestStateLocal
        MaxHalfWindowSize ← AppliedHalfWindowSize
    else if BestStateLocal[1] = BestStateGlobal then
        if AppliedHalfWindowSize > MaxHalfWindowSize then
            BestStateGlobal ← BestStateLocal
            MaxHalfWindowSize ← AppliedHalfWindowSize
        end if
    end if
end for
```

6.7 Validation

In this section, we perform a series of real-world experiments E^P to validate the results generated by the DTs. In the previous section, we determined $s^{\text{pre}}_{o,\text{best}}$ and $s^{\text{pre}}_{o,\text{worst}}$ for each virtual object o^V. For each of these pre-transfer states, we determined $\overline{y}^V$ as the mean result of binary object transfer experiments E^V based on 50 repetitions with uniquely sampled docking errors using our synthetic generator. Since each repetition is statistically independent, we can treat $\overline{y}^V$ as an estimate of the probability of success ($\hat{p}^V = \overline{y}^V$) and the repeated object transfer experiments as Bernoulli trials, a series of success–failure experiments. Note that $\hat{p}^V$ is only an estimate of the true probability p^V. One can calculate a confidence interval which encloses p^V with a given confidence level. We use the Clopper–Pearson interval (Clopper and Pearson 1934) which is known to be a

rather conservative method of calculating the confidence interval for a binomial distribution (Orawo 2021). In Figure 6.22 we have plotted the values $\hat{p}^V$ and the calculated 95% confidence intervals for $s^{\text{pre}}_{o,\text{best}}$ for each of the virtual objects o^V. We can observe that most of the determined $s^{\text{pre}}_{o,\text{best}}$ have a value of $\hat{p}^V \simeq 1$ indicating that the object can be virtually transferred with a high probability of success given the pre-transfer state and is not prone to various docking errors. We can observe that $s^{\text{pre}}_{o,\text{best}}$ of o^V_{36} has only a medium high probability of success ($\hat{p}^V = 0.52$) suggesting that the object is generally difficult to transfer.

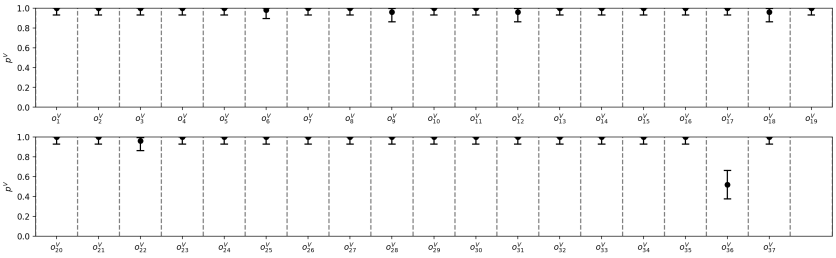


Figure 6.22: 95% confidence intervals including the estimated probability of success $\hat{p}^V$ for $s^{\text{pre}}_{o,\text{best}}$ for each o^V.

Figure 6.23 shows $\hat{p}^V$ and the 95% confidence intervals for $s^{\text{pre}}_{o,\text{worst}}$ of each o^V. For the majority of objects, $s^{\text{pre}}_{o,\text{worst}}$ has a probability of success $\hat{p}^V \simeq 0$. This indicates that in this given state, the object cannot be successfully transferred by the DTs. Some objects, e.g. $o^V_{19}, o^V_{23}, o^V_{25}$ or o^V_{28} have considerable higher probabilities for $s^{\text{pre}}_{o,\text{worst}}$. The object o^V_{28} has a probability of $\hat{p}^V = 1.0$ for its worst state. This indicates that the object could be successfully be transferred by the DTs for all pre-transfer states ${}^{E}\mathcal{S}^{\text{pre}}_{o}$, see Figure A.1ab for the corresponding full-state plot.

To be able to validate the predictions by the DTs, we conducted an additional set of real-world experiments, which we refer to as *Validation Experiments*. In these experiments, see Table 6.6, we physically transfer each object $o^P \in \mathcal{O}^P$ by setting both $s^{\text{pre}}_{o,\text{best}}$ and $s^{\text{pre}}_{o,\text{worst}}$ as determined in the virtual experiments. Figure 6.24 summarises the combined results for $s^{\text{pre}}_{o,\text{best}}$. The blue confidence intervals

Table 6.6: Experiment description: *Validation Experiments*.

Name:	*Validation Experiments*	**Number of runs:**	20
Type:	physical (E^P)	**Number of experiments per run:**	74 (37 for *best* state, 37 for *worst* state)
Transport Module(s):	TM_1^P	**Total number of experiments:**	1480
Transfer Unit(s):	$TU^P \in \mathcal{TU}^P$		
Objects:	Full physical set $\mathcal{O}^P$		

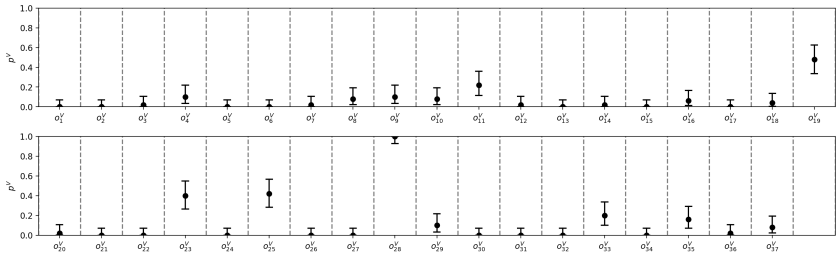


Figure 6.23: 95% confidence intervals including the estimated probability of success $\hat{p}^V$ for $s^{\text{pre}}_{o,\text{worst}}$ for each o^V.

refer to the *Validation Experiments* results. Due to limited set of available physical assets and the long duration of a single physical experiment E^P ($\approx$ 3min), we only repeated each run 20 times which subsequently leads to wider confidence intervals compared to the virtual full-state experiments (50 repetitions). To compare the confidence intervals, we use the conservative *Newcombe Hybrid Score* confidence interval (Newcombe 1998) for the difference $(\hat{p}^V - \hat{p}^P)$. This is one of the recommended methods by (Fagerland et al. 2015) for comparing the results of two independent Bernoulli experiments. For (12/37) objects a significant difference between the confidence intervals exists. In Figure 6.24, we marked these objects with a red background color in their respective columns.

Figure 6.25 shows the combined results for $s^{\text{pre}}_{o,\text{worst}}$. Here (17/37) objects have significant different confidence intervals for the virtual and the physical results. The overall results suggest some sources of error that prevent the DTs from estimating the transfer probabilities more accurately. These errors can be either:

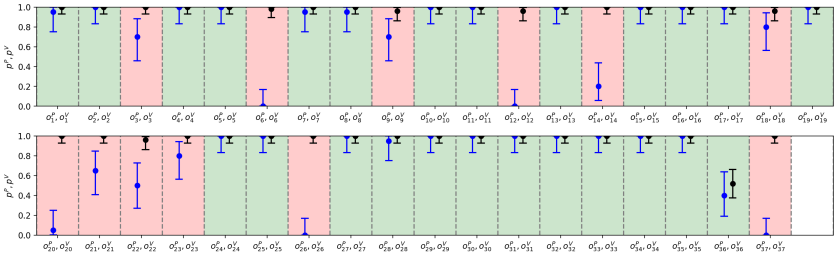


Figure 6.24: Confidence interval comparison between virtual and physical results for $s^{\text{pre}}_{o,\text{best}}$.

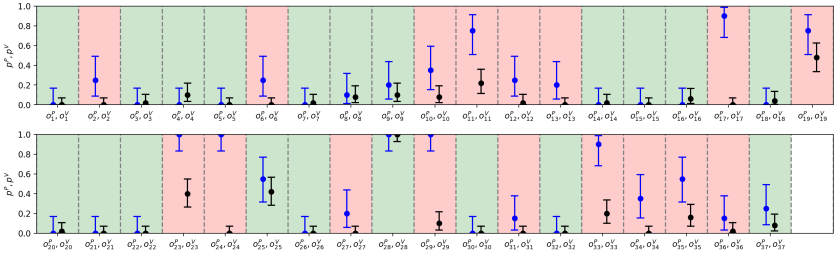


Figure 6.25: Confidence interval comparison between virtual and physical results for $s^{\text{pre}}_{o,\text{worst}}$.

- asset-related, meaning that the virtual models of the transport module, transfer unit or the considered objects do not sufficiently model the real system,
- or physics-engine related, meaning that the engine is either not sufficiently calibrated or in general not sufficient enough to capture the physical interaction during the object transfers.

Defining the exact reason for each individual case is a difficult task as many errors may overlap and lead to the individual results. In the following, we give three implications for sources of error using examples from the *Validation experiments*:

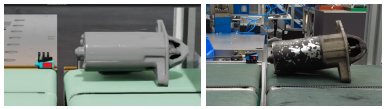
- **Mass Distribution Approximation**: The virtual object models are all approximated with a uniform mass distribution across the collision volume.

This approximation does not accurately reflect the true mass distribution of the objects, leading to incorrect calculations of forces and during the transfer process. For the object transfer example, the real objects may have different contact points during the process compared to the virtual one. For some objects, we can observe this difference in the distribution of mass already during the pre-transfer state. Figure 6.26 visualises two cases for o_{14} and o_{22}.

- **Unstable Physics Simulation**: We are experiencing issues with the physics engine during some object transfers, where objects are unexpectedly catapulted into the air. This problem occurs when the physics simulation fails to correctly handle the contact transition, causing objects to experience excessive and unintended forces, disrupting the expected behaviour and stability of the simulation. We have observed this issue for $s^{\text{pre}}_{o,\text{worst}}$ for objects o^V_6, o^V_{17} and o^V_{22}. Figure 6.26 shows two examples for o^V_6 and o^V_{17}.
- **Inaccurate Collision Shapes**: We have observed two cases, where the virtual (scanned) model of the starter motor core is not accurate. Figure 6.27 shows both cases. o^P_{18} has an elastic part, which is rigid for the scanned virtual model o^V_{18}, see Figure 6.27a. For $s^{\text{pre}}_{o,\text{best}}$, this affects the contact points during the object transfer process. o^P_{24} has a cable tie that was included in the scan is rigid in the collision model.

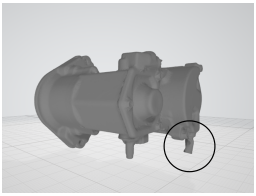
In summary, while the estimates of the DTs are promising, additional modelling efforts may be required to further improve the prediction quality. Future studies could explore the effect of additional object knowledge (e.g. about the correct mass distribution) on the estimate results.

(a) o^V_{14} and o^P_{14} in $s^{\text{pre}}_{o,\text{best}}$

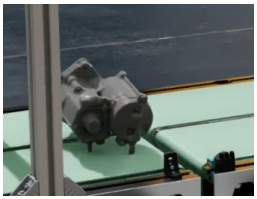


(b) o^V_{22} and o^P_{22} in $s^{\text{pre}}_{o,\text{best}}$

Figure 6.26: Examples for an inaccurate mass distribution: The virtual models have different contact points compared to their physical twins.

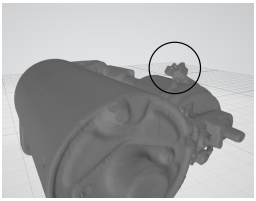


(a) Marked elastic part, which is rigid for o_{18}^{V}.

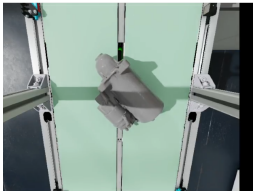


(b) o_{18}^{V} in contact with this part for $s_{o,\text{best}}^{\text{pre}}$

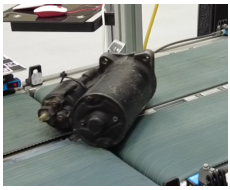
(c) o_{18}^{P} in $s_{o,\text{best}}^{\text{pre}}$.



(d) o_{24}^{V} includes flexible cable tie.

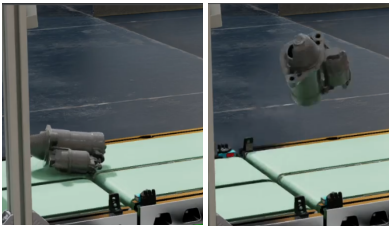


(e) o_{24}^{V} fails for $s_{o,\text{worst}}^{\text{pre}}$ due to collision with the marked part.



(f) o_{24}^{P} can be transferred successfully for $s_{o,\text{worst}}^{\text{pre}}$

Figure 6.27: Examples for inaccurate virtual object models: o_{18}^{V} and o_{22}^{V} both include rigid parts, which for the physical twin are flexible.



(a) o_{6}^{V} in $s_{o,\text{worst}}^{\text{pre}}$

(b) o_{17}^{V} in $s_{o,\text{worst}}^{\text{pre}}$

Figure 6.28: Examples for an unstable physics simulation: The motor models are launching into the air after entering the conveyor gap.

7 Conclusion

This chapter summaries the most important results of this thesis and presents an outlook on further research topics.

7.1 Summary

Remanufacturing, one of the end-of-life strategies, is often considered to be the most effective strategy for saving materials and energy. At present, the high level of uncertainty in remanufacturing, in addition to economic reasons, hinders its wider adoption across various industries. Industry 4.0 methods are frequently proposed as a solution to reduce this uncertainty in remanufacturing processes. As one of the key technologies of Industry 4.0, *Digital Twins* (DTs) have received a lot of research attention over the last decade. We define a DT as a virtual model of a physical twin, coupled with an analytics engine that influences the decision- making process of the physical twin through bidirectional communication. Based on a structured literature review, we found that in the remanufacturing and disassembly automation domain, the majority of research on DTs focuses on a life cycle and supply chain perspective. There are only few contributions at the shop floor level. In addition, there is little applied research on DTs in remanufacturing that has been validated in a real-world application. This is commonly referred to as a current research gap in the remanufacturing community. This thesis contributes in closing this gap by providing a real-world implementation and an extensive application study to validate the capabilities of the DTs. The four contributions of this thesis are summarised below.

Digtial Twin Architecture

The proposed *Digital Twin Architecture* is based on the vision of a fully autonomous, intelligent and self-reconfiguring factory in which autonomous assets form an interconnected *Cyber-Physical System* (CPS) to solve collaborative tasks. The architecture is non-hierarchic with all physical and logical assets existing on the same level. All assets expose their virtual representation that allow to establish communication channels in between them. We define five different types of physical assets based on their ability to synchronise their states and provided external control interfaces. Virtual Models, a subset of the logical assets are the main building block of the DTs. On the cyber layer, the virtual models look identical to their physical counterpart. Finally, a knowledge graph on the descriptive layer acts as the semantic backbone of the architecture.

Formal Object Transfer System

We present a formal description model for the simplest form of an *Object Transfer System* (OTS). In this form, an OTS consists of three assets, an asset that possesses the object to be transferred, an asset that is intended to possess the object after the transfer process, and the object to be transferred itself. The core of the model evolves around a state description of the object to be transferred. During a transfer, the state of the object changes over time. We therefore model the object transfer process as a state transition function. We define several important sets of states that are used to control the object transfer process. In addition, we have provided three examples to show that the model is valid in different object transfer scenarios.

Implementation of an Object Transfer System in a Remanufacturing Demonstration Environment

We present details of an implemented CPS for realising object transfers as part of the *AgiProbot* demonstration system. The system includes three types of

physical assets, the transport unit, a vision-enhanced AMR, the transfer units, stationary transfer points at each station, and a set of starter motor cores, which we consider to be the set of objects to be transferred. The implementation further includes three logical assets, namely the *Transport Module System*, *PropS* and the *Experiment Manager*. For each physical asset, we have implemented virtual models that complete the overall CPS based on the proposed *Digital Twin Architecture*. Through interaction, these assets enable the flexible material flow in the *AgiProbot* demonstration system. The transport module handles transport operations that first require the execution of a global navigation and docking process to a target transfer unit. Once docked, the transport module forms an OTS with the transfer unit. The vision system of the transport module is used to detect and track various objects and to control the object transfer process. Challenges in the implemented system arise from the accuracy of the docking procedure, the discrimination of different objects and a physical gap between the conveyor belts of the docked transport module and the stationary transfer unit, which can result in objects without a container not being able to be transferred.

7.1.0.1 Digital Twin Application Study

We use the implemented CPS in a series of experiments to test the ability of the implemented digital twins to predict the probability of success of an object transfer process for each of the considered starter motor cores, given different pre-transfer states. In a five-step process, we first perform a limited set of real-world experiments using only four different starter motor cores. We use these results to calibrate our virtual models and to train a synthetic generator which we further use to sample synthetically generated docking pose errors. The virtual models achieve a $F_{0.5}$ score of ≈ 0.84 compared to the results of the calibration experiments. We then conduct a large series of virtual experiments, which we use to estimate the probability of success for each pre-transfer state for each of the starter motor cores. Based on the results, we propose a method to determine the arguably *best* and *worst* pre-transfer states. Finally, we perform an additional set of real-world experiments to validate the estimates of the DTs. For

the best states, (12/37) and for the worst state, (17/37) objects show significant differences between the virtual and the real-world results based on the calculated confidence intervals. We have discussed several sources of error, including the mass distribution approximation, unstabilities in the physics simulation and inaccurate collision shapes. Further experiments are needed to investigate the effect of increased modeling effort on the estimation results.

7.2 Outlook

Based on the results, we identify future research perspectives that go beyond the scope of this thesis but which we believe should be further explored.

We investigated the predictions of the DTs of an object transfer system detached from its integration in a higher-level system, such as a production control system that schedules tasks and allocates resources. Further research based on a (simulated) production control system that integrates these predictions would allow for a quantification of improvements in terms of factory-level *Key Performance Indicators* (KPIs).

In the proposed DT architecture, we have only briefly mentioned the use and application of a knowledge graph as the semantic backbone of the CPS. Detailed research should therefore be carried out on its implementation as a runtime system on the shop floor and its overall capabilities for inferring information on about different concepts and relations. This includes the plug-and-play capabilities of the CPS and further research on asset discovery and reasoning about each asset's capabilities.

In our application study, we investigated the implementation of a single, specific object transfer system that is capable of transferring containerless objects using conveyor belts. Further research can be conducted on the implementation of the proposed concepts for different object transfer scenarios. We have discussed a closely related scenario in (Behrendt et al. 2023), where we describe the reconfiguration approach of the fluid automated station (Station C) as part of the

AgiProbot demonstration system. The material handling operations performed by a collaborative robot could also be described by the formal description model and embedded in the DT-based CPS. The pick-and-place operations in this case are also probabilistic and a similar approach to predicting the probability of success of these operations is conceivable.

A Appendix

A.1 Calibration Experiments

Table A.1: Quantitative calibration experiment results

			Pre-transfer state s^{pre}				${}_e\tilde{\xi}$			
ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta$ [°]	A $[cm^2]$
0	TU_D^P	o_1^P	-11	front	right	1	-8.55	2.84	1.44	13.41
1	TU_A^P	o_1^P	-4	front	left	0	-6.41	9.97	2.3	28.09
2	TU_C^P	o_1^P	2	front	left	0	-0.64	-15.38	0.93	6.96
3	TU_D^P	o_1^P	-140	front	both	1	-13.42	10.9	1.62	33.15
4	TU_C^P	o_1^P	-173	front	right	0	-5.07	-17.31	1.87	3.21
5	TU_D^P	o_1^P	29	front	both	1	-15.09	5.61	1.5	37.53
6	TU_A^P	o_1^P	34	back	both	0	-11.21	7.29	1.55	27.78
7	TU_D^P	o_1^P	66	back	both	0	-11.76	8.89	1.77	29.05
8	TU_B^P	o_1^P	23	front	left	1	-9.61	-17.85	0.44	26.61
9	TU_A^P	o_1^P	89	back	both	0	-7.47	9.89	2.09	18.23
10	TU_A^P	o_1^P	16	front	left	1	-9.0	9.57	1.57	30.7
11	TU_A^P	o_1^P	27	front	both	0	-8.65	5.19	1.71	21.42
12	TU_D^P	o_1^P	132	front	both	1	-11.09	12.36	1.64	27.27
13	TU_D^P	o_1^P	100	back	both	0	-11.01	5.85	1.06	27.39
14	TU_D^P	o_1^P	-52	back	both	0	-10.85	7.75	1.56	26.85
15	TU_A^P	o_1^P	-132	front	both	1	-5.91	12.01	2.34	14.16
16	TU_B^P	o_1^P	-108	front	both	1	-5.23	-11.26	1.26	13.39
17	TU_A^P	o_1^P	41	back	both	0	-6.4	-4.1	1.49	16.14
18	TU_A^P	o_1^P	53	front	both	1	-9.39	0.73	1.72	23.43
19	TU_D^P	o_1^P	135	front	both	1	-9.16	8.03	1.22	22.68
20	TU_B^P	o_1^P	23	back	right	0	-5.5	-12.55	1.21	7.48

Continued on next page

Table A.1 – Continued from previous page

ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta[^\circ]$	A $[cm^2]$
21	TU_A^P	o_1^P	37	back	both	0	-7.83	6.06	1.96	19.3
22	TU_D^P	o_1^P	-8	front	right	0	-7.99	7.17	1.68	10.55
23	TU_A^P	o_1^P	157	back	left	0	-7.82	2.51	1.66	28.52
24	TU_C^P	o_1^P	-126	front	both	1	-1.47	-21.94	1.84	4.54
25	TU_C^P	o_1^P	-79	front	both	1	-2.18	-29.42	0.28	5.62
26	TU_B^P	o_1^P	147	back	both	0	-5.19	-7.19	1.9	13.27
27	TU_C^P	o_1^P	163	front	right	1	-0.9	-23.16	0.9	2.21
28	TU_C^P	o_1^P	-96	back	both	1	-0.16	-27.01	0.63	0.76
29	TU_C^P	o_1^P	54	front	both	1	-1.16	-27.37	0.01	2.92
30	TU_C^P	o_1^P	145	front	both	1	-0.57	-21.36	1.1	1.93
31	TU_B^P	o_1^P	-38	back	both	0	-5.58	-14.3	1.55	14.44
32	TU_C^P	o_1^P	8	back	right	1	-1.02	-13.53	0.87	1.93
33	TU_C^P	o_1^P	171	back	left	0	-0.15	-17.48	0.39	2.63
34	TU_C^P	o_1^P	-24	front	left	1	-0.32	-15.46	1.08	7.08
35	TU_C^P	o_1^P	93	front	both	1	-1.94	-12.98	1.0	5.13
36	TU_B^P	o_1^P	141	front	both	1	-6.57	-11.35	1.88	16.89
37	TU_D^P	o_1^P	-91	back	both	0	-9.95	3.09	1.1	24.8
38	TU_C^P	o_1^P	91	back	both	0	-2.59	-8.16	2.14	6.84
39	TU_B^P	o_1^P	1	front	right	1	-4.52	-7.49	1.6	2.85
40	TU_A^P	o_1^P	-40	front	both	1	-7.72	5.62	1.67	19.09
41	TU_B^P	o_1^P	10	back	right	0	-4.77	-9.75	1.34	4.9
42	TU_C^P	o_1^P	-127	back	both	1	-2.13	-16.1	0.84	5.62
43	TU_D^P	o_1^P	22	back	left	0	-8.94	5.05	1.23	28.93
44	TU_B^P	o_1^P	-144	front	both	1	-4.52	-4.12	1.98	11.47
45	TU_C^P	o_1^P	167	front	right	1	-0.5	-9.22	1.53	6.8
46	TU_D^P	o_1^P	-139	back	both	0	-9.2	2.82	1.52	22.9
47	TU_C^P	o_1^P	-30	front	both	1	-1.95	-17.02	1.22	5.34
48	TU_D^P	o_1^P	-77	back	both	0	-8.83	5.19	1.77	21.86
49	TU_B^P	o_1^P	-102	front	both	1	-5.8	-5.7	1.68	14.69
50	TU_A^P	o_1^P	-2	front	left	0	-7.78	3.75	1.89	29.57
51	TU_B^P	o_1^P	-25	back	left	0	-3.26	-1.92	1.87	18.41
52	TU_B^P	o_1^P	-156	front	left	1	-6.44	-16.62	-0.23	14.76
53	TU_A^P	o_1^P	-89	back	both	0	-9.74	4.6	0.71	24.29
54	TU_D^P	o_{32}^P	117	back	both	1	-10.9	2.96	0.55	27.22
55	TU_A^P	o_{32}^P	-59	front	both	1	-8.46	10.98	1.72	20.74
56	TU_D^P	o_{32}^P	129	front	both	1	-11.42	8.01	0.98	28.37

Continued on next page

Table A.1 – Continued from previous page

ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta[^\circ]$	A $[cm^2]$
57	TU_B^P	o_{32}^P	4	front	left	0	-9.3	-13.98	0.36	25.32
58	TU_D^P	o_{32}^P	-132	front	both	1	-13.07	-0.28	0.94	32.67
59	TU_A^P	o_{32}^P	56	back	both	1	-9.95	4.89	1.27	24.73
60	TU_C^P	o_{32}^P	-76	front	both	1	-4.79	-7.65	1.63	12.24
61	TU_C^P	o_{32}^P	-77	back	both	1	-3.49	-15.39	-0.1	8.68
62	TU_A^P	o_{32}^P	162	front	left	0	-9.74	3.1	1.03	29.88
63	TU_B^P	o_{32}^P	-57	back	both	1	-6.6	-9.91	2.04	16.94
64	TU_A^P	o_{32}^P	-69	back	both	1	-6.72	-1.55	1.79	16.86
65	TU_D^P	o_{32}^P	-128	back	both	1	-9.81	9.93	0.98	24.32
66	TU_C^P	o_{32}^P	24	front	left	1	-1.05	-20.15	0.52	5.71
67	TU_B^P	o_{32}^P	-30	back	both	1	-7.06	-14.86	2.2	18.36
68	TU_B^P	o_{32}^P	-2	front	left	0	-6.39	-3.76	1.54	24.52
69	TU_D^P	o_{32}^P	7	back	left	1	-10.25	2.7	1.36	32.97
70	TU_C^P	o_{32}^P	-167	front	left	0	-1.51	-6.23	0.32	5.59
71	TU_C^P	o_{32}^P	167	front	right	0	-2.05	-12.57	0.45	2.79
72	TU_A^P	o_{32}^P	-147	back	both	1	-8.37	2.03	1.63	20.85
73	TU_B^P	o_{32}^P	-67	back	both	1	-6.15	-8.13	1.72	15.69
74	TU_A^P	o_{32}^P	28	front	both	1	-7.61	5.51	2.25	18.74
75	TU_B^P	o_{32}^P	118	back	both	1	-6.52	-7.25	1.22	16.49
76	TU_B^P	o_{32}^P	160	front	right	0	-6.28	-2.61	1.66	6.72
77	TU_A^P	o_{32}^P	-110	back	both	1	-7.96	0.37	1.66	19.89
78	TU_B^P	o_{32}^P	-132	back	both	1	-8.92	-12.19	1.46	22.69
79	TU_D^P	o_{32}^P	129	back	both	1	-14.32	21.22	1.11	35.28
80	TU_C^P	o_{32}^P	59	front	both	1	-5.71	-22.29	0.78	14.66
81	TU_B^P	o_{32}^P	-144	back	both	1	-8.65	-6.12	1.39	21.82
82	TU_A^P	o_{32}^P	1	front	right	0	-10.4	16.87	1.99	14.41
83	TU_D^P	o_{32}^P	-94	front	both	1	-12.13	12.13	1.29	29.97
84	TU_C^P	o_{32}^P	168	front	left	0	-6.33	-5.93	1.8	25.89
85	TU_D^P	o_{32}^P	154	front	both	0	-12.12	21.9	1.49	29.57
86	TU_A^P	o_{32}^P	91	front	both	0	-11.28	15.39	1.75	27.61
87	TU_D^P	o_{32}^P	69	back	both	1	-14.52	17.75	0.92	35.95
88	TU_C^P	o_{32}^P	82	back	both	1	-4.06	-14.66	2.01	10.79
89	TU_C^P	o_{32}^P	-164	front	left	0	-7.73	-4.52	2.38	32.51
90	TU_D^P	o_{32}^P	-144	front	both	0	-10.17	9.77	1.14	25.18
91	TU_B^P	o_{32}^P	-150	front	both	1	-6.57	-6.3	1.72	16.66
92	TU_B^P	o_{32}^P	167	front	left	0	-6.96	-13.34	0.78	21.87

Continued on next page

Table A.1 – Continued from previous page

ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta[^\circ]$	A $[cm^2]$
93	TU_A^P	o_{32}^P	89	front	both	1	-7.19	6.79	1.93	17.69
94	TU_A^P	o_{32}^P	-170	front	right	0	-7.09	6.36	1.62	8.69
95	TU_D^P	o_{32}^P	77	front	both	1	-9.76	3.11	0.57	24.37
96	TU_A^P	o_{32}^P	138	front	both	1	-6.54	11.78	2.03	15.83
97	TU_C^P	o_{32}^P	-64	front	both	1	-3.72	-13.38	0.92	9.56
98	TU_A^P	o_{18}^P	-18	back	right	1	-5.64	6.85	1.33	6.64
99	TU_B^P	o_{18}^P	-114	front	both	0	-6.38	-6.66	1.94	16.22
100	TU_B^P	o_{18}^P	30	back	both	0	-5.62	-17.15	0.82	14.35
101	TU_B^P	o_{18}^P	82	back	both	0	-5.4	-8.13	1.51	13.77
102	TU_C^P	o_{18}^P	-136	front	both	1	-1.09	-11.18	1.48	3.07
103	TU_D^P	o_{18}^P	8	front	left	0	-9.88	9.36	1.78	34.05
104	TU_C^P	o_{18}^P	-89	front	both	1	-4.29	-14.96	0.7	10.95
105	TU_C^P	o_{18}^P	-29	back	both	1	-6.28	-2.96	1.48	15.79
106	TU_A^P	o_{18}^P	-102	front	both	0	-11.6	2.9	1.35	28.92
107	TU_C^P	o_{18}^P	-74	back	both	1	-3.39	-6.63	0.57	8.55
108	TU_D^P	o_{18}^P	-138	back	both	0	-13.53	10.39	0.71	33.65
109	TU_C^P	o_{18}^P	176	front	left	1	-4.88	-2.59	0.94	17.38
110	TU_D^P	o_{18}^P	-137	back	both	0	-11.15	9.42	0.81	27.72
111	TU_A^P	o_{18}^P	93	front	both	1	-9.56	-0.12	0.79	23.89
112	TU_C^P	o_{18}^P	-114	back	both	0	-2.88	-19.14	0.21	7.28
113	TU_D^P	o_{18}^P	52	front	both	1	-11.47	11.24	0.55	28.53
114	TU_A^P	o_{18}^P	91	front	both	0	-8.78	2.96	0.68	21.9
115	TU_D^P	o_{18}^P	-173	back	right	1	-11.73	9.81	0.57	26.11
116	TU_B^P	o_{18}^P	-71	back	both	0	-9.01	-12.01	-0.58	22.36
117	TU_D^P	o_{18}^P	-174	front	left	1	-11.2	12.58	0.77	31.99
118	TU_D^P	o_{18}^P	98	front	both	1	-13.15	8.31	0.39	32.81
119	TU_A^P	o_{18}^P	33	front	both	0	-21.43	1.34	0.79	53.55
120	TU_C^P	o_{18}^P	-95	back	both	1	-4.17	-5.96	0.92	10.55
121	TU_A^P	o_{18}^P	-176	back	left	0	-9.85	5.48	0.74	28.57
122	TU_A^P	o_{18}^P	129	back	both	0	-9.5	5.61	1.83	23.51
123	TU_A^P	o_{18}^P	37	front	both	1	-9.07	2.74	1.1	22.61
124	TU_B^P	o_{18}^P	24	back	right	1	-9.26	-10.55	1.05	17.67
125	TU_B^P	o_{18}^P	103	back	both	0	-9.14	-5.0	0.67	22.93
126	TU_B^P	o_{18}^P	134	back	both	1	-9.09	-16.64	-0.8	22.43
127	TU_B^P	o_{18}^P	57	back	both	0	-8.39	-10.66	1.12	21.24
128	TU_D^P	o_{18}^P	127	front	both	1	-11.61	6.59	0.06	29.02

Continued on next page

Table A.1 – Continued from previous page

ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta[^\circ]$	A $[cm^2]$
129	TU_B^P	o_{18}^P	-85	back	both	0	-5.87	-13.08	-0.11	14.64
130	TU_D^P	o_{18}^P	-145	back	both	1	-10.07	14.44	0.82	24.91
131	TU_B^P	o_{18}^P	-149	back	both	0	-7.68	0.91	1.37	19.16
132	TU_C^P	o_{18}^P	138	back	both	1	-6.13	-4.47	1.47	15.47
133	TU_C^P	o_{18}^P	-61	front	both	1	-4.01	-17.37	-0.3	9.92
134	TU_C^P	o_{18}^P	163	front	left	1	-5.12	-7.56	1.0	18.42
135	TU_C^P	o_{18}^P	-100	back	both	1	-6.13	-4.08	1.51	15.45
136	TU_C^P	o_{18}^P	-121	front	both	1	-4.16	-9.36	0.59	10.52
137	TU_C^P	o_{18}^P	94	front	both	1	-4.72	-5.45	1.07	11.91
138	TU_B^P	o_{18}^P	-63	back	both	0	-7.37	-6.09	1.6	18.63
139	TU_A^P	o_{18}^P	19	front	left	1	-8.63	7.89	1.33	28.58
140	TU_C^P	o_{18}^P	-79	front	both	1	-5.69	-1.56	1.73	14.29
141	TU_B^P	o_{18}^P	-59	front	both	1	-11.46	-8.23	1.86	28.98
142	TU_C^P	o_{18}^P	-16	back	right	1	-6.01	-12.68	1.85	5.42
143	TU_C^P	o_{18}^P	61	front	both	1	-5.3	-18.36	0.46	13.44
144	TU_B^P	o_{18}^P	70	front	both	1	-10.01	-7.38	1.54	25.27
145	TU_B^P	o_{18}^P	-140	back	both	0	-11.39	-26.23	0.8	28.92
146	TU_C^P	o_{18}^P	-3	front	right	1	-5.76	-6.36	1.6	5.88
147	TU_D^P	o_{18}^P	80	front	both	1	-13.08	8.13	0.39	32.62
148	TU_A^P	o_{18}^P	-91	back	both	0	-11.18	5.07	1.24	27.8
149	TU_B^P	o_{18}^P	56	front	both	1	-8.12	-6.27	1.65	20.52
150	TU_B^P	o_{23}^P	-28	back	both	1	-8.91	-26.49	0.17	22.37
151	TU_A^P	o_{23}^P	88	front	both	0	-9.51	8.4	0.89	23.61
152	TU_B^P	o_{23}^P	115	back	both	1	-6.39	-27.46	-0.17	15.88
153	TU_A^P	o_{23}^P	164	front	left	1	-8.87	3.12	1.24	28.87
154	TU_A^P	o_{23}^P	42	front	both	1	-4.35	9.93	2.62	10.3
155	TU_D^P	o_{23}^P	-38	back	both	1	-7.44	4.33	1.64	18.44
156	TU_A^P	o_{23}^P	-21	front	left	1	-4.45	3.0	2.09	22.37
157	TU_A^P	o_{23}^P	-179	back	right	1	-3.88	-1.0	2.25	2.55
158	TU_C^P	o_{23}^P	170	front	right	1	1.59	-9.33	1.82	13.53
159	TU_A^P	o_{23}^P	-75	back	both	1	-4.04	7.93	2.15	9.73
160	TU_B^P	o_{23}^P	8	front	left	1	-4.42	-11.6	1.55	19.86
161	TU_D^P	o_{23}^P	106	front	both	1	-6.07	3.27	1.53	15.06
162	TU_B^P	o_{23}^P	121	back	both	1	-3.59	-13.88	1.48	9.42
163	TU_A^P	o_{23}^P	-60	back	both	1	-4.64	0.56	2.16	11.56
164	TU_A^P	o_{23}^P	170	back	right	1	-4.09	2.49	2.54	3.77

Continued on next page

Table A.1 – Continued from previous page

ID-E	TU^P	o^P	θ	f	c	y	${}_e\tilde{\xi}_x$ [mm]	${}_e\tilde{\xi}_y$ [mm]	${}_e\tilde{\xi}_\theta[^\circ]$	A $[cm^2]$
165	TU_A^P	o_{23}^P	85	back	both	1	-5.15	7.55	3.09	12.36
166	TU_C^P	o_{23}^P	-144	front	both	1	-0.74	-12.16	2.58	2.54
167	TU_A^P	o_{23}^P	58	back	both	1	-4.57	0.39	2.16	11.41
168	TU_B^P	o_{23}^P	40	front	both	1	-4.32	-16.69	0.99	11.16
169	TU_C^P	o_{23}^P	-73	back	both	1	0.67	-15.12	1.46	1.2
170	TU_A^P	o_{23}^P	-86	front	both	0	-8.64	3.67	1.4	21.48
171	TU_A^P	o_{23}^P	80	back	both	1	-8.8	2.29	0.97	21.95
172	TU_A^P	o_{23}^P	-71	back	both	1	-7.47	4.25	1.33	18.54
173	TU_B^P	o_{23}^P	13	front	right	1	-8.33	-11.09	0.35	19.0
174	TU_A^P	o_{23}^P	11	back	left	1	-8.62	0.69	0.83	26.08
175	TU_D^P	o_{23}^P	-68	front	both	1	-11.62	6.29	0.59	28.96
176	TU_A^P	o_{23}^P	-83	back	both	1	-9.49	3.27	0.95	23.66
177	TU_B^P	o_{23}^P	-99	back	both	1	-7.76	-10.18	0.66	19.55
178	TU_D^P	o_{23}^P	-34	front	both	0	-9.48	7.3	0.75	23.57
179	TU_A^P	o_{23}^P	-59	back	both	1	-7.3	9.63	1.08	18.02
180	TU_B^P	o_{23}^P	112	back	both	1	-7.71	-8.93	1.02	19.48
181	TU_B^P	o_{23}^P	166	back	left	1	-8.94	-9.92	0.81	26.98
182	TU_A^P	o_{23}^P	-77	front	both	1	-8.18	1.12	1.02	20.43
183	TU_B^P	o_{23}^P	136	front	both	0	-7.68	-14.71	0.08	19.23
184	TU_C^P	o_{23}^P	78	front	both	1	-4.29	-3.24	1.35	10.81
185	TU_C^P	o_{23}^P	-35	back	both	1	-3.7	-11.28	0.94	9.48
186	TU_A^P	o_{23}^P	-36	front	both	1	-9.21	2.92	1.12	22.94
187	TU_D^P	o_{23}^P	-53	front	both	1	-10.53	6.8	0.57	26.24
188	TU_B^P	o_{23}^P	109	back	both	1	-7.71	-2.41	1.48	19.36
189	TU_A^P	o_{23}^P	-1	back	right	1	-8.72	8.41	2.29	8.91
190	TU_A^P	o_{23}^P	33	front	both	0	-8.67	4.02	1.72	21.51
191	TU_A^P	o_{23}^P	-45	back	both	1	-7.75	10.91	1.49	19.01
192	TU_A^P	o_{23}^P	-74	front	both	1	-7.59	10.44	1.12	18.72
193	TU_A^P	o_{23}^P	97	back	both	1	-8.38	6.36	1.53	20.74
194	TU_D^P	o_{23}^P	-13	back	left	1	-10.99	8.9	0.84	31.9
195	TU_D^P	o_{23}^P	-175	back	right	1	-10.43	8.44	0.58	22.83
196	TU_C^P	o_{23}^P	-119	front	both	1	-3.07	-12.34	0.69	7.86
197	TU_C^P	o_{23}^P	162	front	right	1	-5.45	-2.77	1.65	4.73
198	TU_D^P	o_{23}^P	4	back	left	1	-10.58	5.99	0.95	31.49
199	TU_D^P	o_{23}^P	-3	front	left	1	-11.12	8.58	0.95	32.8

A.2 Evaluation Metrics

Table A.2: Confusion Matrix.

		Predicted condition	
	Total population = P + N	Predicted Positive (PP)	Predicted Negative (PN)
Actual condition	Positive (P)	True Positive (TP)	False Nevgative (FN)
	Negative (N)	False Positive (FP)	True negative (TN)

$$\text{precision} = \frac{TP}{TP + FP} = \frac{TP}{PP} \tag{A.1}$$

$$\text{recall} = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{A.2}$$

$$\text{accuracy} = \frac{TP + TN}{P + N} \tag{A.3}$$

$$\text{F}_{0.5} = (1 + 0.5^2) * \frac{\text{precision} \cdot \text{recall}}{0.5^2 \cdot \text{precision} + \text{recall}} \tag{A.4}$$

A.3 Parameter Sets

Table A.3: Constant simulation parameters.

CCD	Maximum Position Iteration Count	Maximum Velocity Iteration Count	Minimum Position Iteration Count	Bounce Threshold [m/s]	Enhanced Determinism	Friction Correlation Distance [m]
True	255	255	1	2.0	True	0.025

Table A.4: Evaluated simulation parameter combinations (ID-S).

ID-S	Time Steps Per Second [1/s]	Collision System	Solver Type	Broad phase Type	GPU Dynamics	Enable Stabilization	Min Velocity Iteration Count	Friction Offset Threshold [m]	Friction Type
0	50	PCM	TGS	MBP	False	True	1	0.04	patch
1	100	PCM	TGS	MBP	False	True	1	0.04	patch
2	150	PCM	TGS	MBP	False	True	1	0.04	patch
3	200	PCM	TGS	MBP	False	True	1	0.04	patch
4	250	PCM	TGS	MBP	False	True	1	0.04	patch
5	150	SAT	TGS	MBP	False	True	1	0.04	patch
6	150	PCM	PGS	MBP	False	True	1	0.04	patch
7	150	PCM	TGS	SAP	False	True	1	0.04	patch
8	150	PCM	TGS	GPU	True	True	1	0.04	patch
9	150	PCM	TGS	MBP	False	False	1	0.04	patch
10	150	PCM	TGS	MBP	False	True	3	0.04	patch
11	150	PCM	TGS	MBP	False	True	5	0.04	patch
12	150	PCM	TGS	MBP	False	True	7	0.04	patch
13	150	PCM	TGS	MBP	False	True	9	0.04	patch
14	150	PCM	TGS	MBP	False	True	1	0.00	patch
15	150	PCM	TGS	MBP	False	True	1	0.02	patch
16	150	PCM	TGS	MBP	False	True	1	0.06	patch
17	150	PCM	TGS	MBP	False	True	1	0.08	patch
18	150	PCM	TGS	MBP	False	True	1	0.10	patch
19	150	PCM	TGS	MBP	False	True	1	0.04	1D
20	150	PCM	TGS	MBP	False	True	1	0.04	2D

Table A.5: Instance parameter combinations (ID-I).

Generated by a Latin hypercube design: ID-I (1-20) and ID-I (21-40) with different parameter ranges: $d_{gap} = (\{5, 20\}, \{8, 13\})$, $\mu_k = (\{0.4, 0.6\}, \{0.43, 0.49\})$

ID-I	d_{gap}[mm]	μ_k
0	15	0.5
1	9.28	0.45508
2	15.3	0.53090
3	16.64	0.59550
4	15.83	0.46831
5	14.13	0.57178
6	13.55	0.47351
7	9.59	0.55231
8	19.08	0.48460
9	17.63	0.52724
10	19.49	0.56974
11	6.62	0.51165
12	12.82	0.50838
13	7.39	0.58748
14	6.37	0.44774
15	8.0	0.40036
16	10.3	0.49670
17	11.11	0.41149
18	5.36	0.54661
19	12.2	0.43865
20	18.25	0.42124
21	9.43	0.44652
22	11.43	0.46927
23	**11.88**	**0.48865**
24	11.61	0.45049
25	11.04	0.48153
26	10.85	0.45205
27	9.53	0.47569
28	12.69	0.45538
29	12.21	0.46817
30	12.83	0.48092
31	8.54	0.46350
32	10.61	0.46251
33	8.8	0.48624
34	8.46	0.44432
35	9.77	0.45901
36	9.0	0.43011
37	10.04	0.43345
38	8.12	0.47398
39	10.4	0.44160
40	12.42	0.43637

A.4 Result Tables

Table A.7: Evaluation of instance parameter combinations, see Table A.5.

ID-I	ID-S	PP	PN	FP	FN	TP	TN	accuracy	precision	recall	$F_{0.5}$
1	2	124.2 ±3.8	75.8 ±3.8	20.0 ±2.7	30.8 ±3.3	104.2 ±3.3	45.0 ±2.7	0.7462 ±0.024	0.8394 ±0.019	0.7719 ±0.025	0.8248 ±0.018
2	2	114.1 ±3.3	85.9 ±3.3	16.8 ±2.5	37.6 ±3.0	97.4 ±3.0	48.2 ±2.5	0.728 ±0.022	0.8533 ±0.02	0.7212 ±0.022	0.823 ±0.018
3	2	124.0 ±4.2	76.0 ±4.2	23.2 ±2.0	34.2 ±3.4	100.8 ±3.4	41.8 ±2.0	0.7127 ±0.019	0.8127 ±0.014	0.7464 ±0.025	0.7984 ±0.013
4	2	94.3 ±3.6	105.7 ±3.6	9.0 ±2.0	49.7 ±3.1	85.3 ±3.1	56.0 ±2.0	0.7067 ±0.019	0.9049 ±0.019	0.6321 ±0.023	0.8327 ±0.017
5	2	134.0 ±3.9	66.0 ±3.9	25.6 ±2.1	26.7 ±2.8	108.3 ±2.8	39.4 ±2.1	0.7385 ±0.016	0.8088 ±0.013	0.8025 ±0.021	0.8074 ±0.011
6	2	109.8 ±4.0	90.2 ±4.0	14.1 ±2.1	39.2 ±3.6	95.8 ±3.6	50.9 ±2.1	0.7335 ±0.021	0.8721 ±0.017	0.7094 ±0.027	0.8336 ±0.017
7	2	150.3 ±3.8	49.7 ±3.8	32.5 ±1.9	17.3 ±2.7	117.7 ±2.7	32.5 ±1.9	0.751 ±0.014	0.7836 ±0.009	0.8721 ±0.02	0.7998 ±0.009
8	2	82.3 ±3.5	117.7 ±3.5	6.8 ±1.8	59.5 ±2.9	75.5 ±2.9	58.2 ±1.8	0.6683 ±0.016	0.9177 ±0.02	0.559 ±0.021	0.813 ±0.016
9	2	103.1 ±3.7	96.9 ±3.7	12.6 ±2.4	44.5 ±3.0	90.5 ±3.0	52.4 ±2.4	0.7148 ±0.02	0.8784 ±0.021	0.6706 ±0.023	0.827 ±0.018
10	2	101.7 ±3.6	98.3 ±3.6	14.6 ±2.0	47.9 ±2.9	87.1 ±2.9	50.4 ±2.0	0.6877 ±0.018	0.8567 ±0.018	0.6454 ±0.022	0.8039 ±0.015
11	2	141.8 ±5.2	58.2 ±5.2	31.7 ±2.3	24.9 ±4.5	110.1 ±4.5	33.3 ±2.3	0.7172 ±0.024	0.7766 ±0.014	0.8156 ±0.033	0.784 ±0.015
12	2	121.3 ±3.6	78.7 ±3.6	17.7 ±2.1	31.3 ±3.0	103.7 ±3.0	47.3 ±2.1	0.755 ±0.019	0.8546 ±0.016	0.7679 ±0.022	0.8356 ±0.014
13	2	161.3 ±3.3	38.7 ±3.3	44.2 ±2.3	17.9 ±2.3	117.1 ±2.3	20.8 ±2.3	0.6893 ±0.016	0.7259 ±0.011	0.8674 ±0.017	0.7504 ±0.01
14	2	121.5 ±3.8	78.5 ±3.8	23.1 ±2.8	36.6 ±2.6	98.4 ±2.6	41.9 ±2.8	0.7018 ±0.019	0.8105 ±0.019	0.7291 ±0.019	0.7927 ±0.016
15	2	103.2 ±3.9	96.8 ±3.9	13.6 ±1.8	45.4 ±3.3	89.6 ±3.3	51.4 ±1.8	0.7048 ±0.018	0.8683 ±0.015	0.6635 ±0.025	0.8176 ±0.014
16	2	131.4 ±3.9	68.6 ±3.9	22.6 ±2.2	26.2 ±3.4	108.8 ±3.4	42.4 ±2.2	0.7563 ±0.021	0.8284 ±0.015	0.8062 ±0.025	0.8237 ±0.014
17	2	99.0 ±4.2	101.0 ±4.2	12.7 ±1.6	48.7 ±3.9	86.3 ±3.9	52.3 ±1.6	0.6927 ±0.021	0.8714 ±0.015	0.639 ±0.029	0.8121 ±0.015
18	2	147.9 ±3.5	52.1 ±3.5	41.0 ±2.4	28.2 ±2.7	106.8 ±2.7	24.0 ±2.4	0.654 ±0.018	0.7226 ±0.013	0.7914 ±0.02	0.7353 ±0.012

Continued on next page

Table A.7 – Continued from previous page

ID-I	ID-S	PP	PN	FP	FN	TP	TN	accuracy	precision	recall	$F_{0.5}$
19	2	106.6 ±3.7	93.4 ±3.7	12.3 ±1.9	40.7 ±3.7	94.3 ±3.7	52.7 ±1.9	0.7352 ±0.023	0.8849 ±0.018	0.6985 ±0.028	0.8399 ±0.018
20	2	69.0 ±4.0	131.0 ±4.0	3.5 ±1.5	69.5 ±3.9	65.5 ±3.9	61.5 ±1.5	0.6352 ±0.022	0.9498 ±0.021	0.4852 ±0.029	0.7966 ±0.022
21	2	120.5 ±3.4	79.5 ±3.4	18.3 ±2.0	32.8 ±2.8	102.2 ±2.8	46.7 ±2.0	0.7447 ±0.018	0.8483 ±0.015	0.7573 ±0.021	0.8283 ±0.013
22	2	116.9 ±3.7	83.1 ±3.7	16.2 ±1.6	34.3 ±3.1	100.7 ±3.1	48.8 ±1.6	0.7473 ±0.017	0.8613 ±0.012	0.7459 ±0.023	0.8353 ±0.012
23	**2**	**123.5** ±4.3	**76.5** ±4.3	**18.0** ±2.8	**29.5** ±3.1	**105.5** ±3.1	**47.0** ±2.8	**0.7627** ±0.021	**0.8548** ±0.019	**0.7815** ±0.023	**0.8389** ±0.017
24	2	111.6 ±3.0	88.4 ±3.0	15.2 ±2.4	38.7 ±2.1	96.3 ±2.1	49.8 ±2.4	0.7305 ±0.017	0.8638 ±0.019	0.7136 ±0.015	0.8288 ±0.015
25	2	123.8 ±4.0	76.2 ±4.0	18.6 ±2.1	29.8 ±3.7	105.2 ±3.7	46.4 ±2.1	0.7582 ±0.022	0.8501 ±0.016	0.7793 ±0.027	0.8347 ±0.016
26	2	111.1 ±3.5	88.9 ±3.5	15.2 ±1.9	39.0 ±3.2	96.0 ±3.2	49.8 ±1.9	0.729 ±0.019	0.8636 ±0.016	0.7109 ±0.023	0.8279 ±0.015
27	2	129.2 ±3.4	70.8 ±3.4	22.3 ±2.4	28.1 ±2.8	106.9 ±2.8	42.7 ±2.4	0.7483 ±0.02	0.8278 ±0.017	0.7921 ±0.021	0.8203 ±0.015
28	2	110.7 ±2.9	89.3 ±2.9	13.9 ±2.0	38.2 ±3.0	96.8 ±3.0	51.1 ±2.0	0.7397 ±0.021	0.8745 ±0.017	0.7173 ±0.022	0.8377 ±0.017
29	2	114.1 ±4.3	85.9 ±4.3	16.1 ±2.1	37.0 ±3.3	98.0 ±3.3	48.9 ±2.1	0.7343 ±0.018	0.8589 ±0.015	0.7259 ±0.025	0.8283 ±0.013
30	2	115.2 ±3.9	84.8 ±3.9	16.0 ±2.0	35.8 ±3.6	99.2 ±3.6	49.0 ±2.0	0.7412 ±0.021	0.8612 ±0.016	0.7351 ±0.026	0.8325 ±0.015
31	2	129.3 ±3.7	70.7 ±3.7	22.6 ±2.2	28.4 ±3.2	106.6 ±3.2	42.4 ±2.2	0.745 ±0.02	0.825 ±0.015	0.7899 ±0.024	0.8176 ±0.014
32	2	119.0 ±4.6	81.0 ±4.6	17.6 ±2.1	33.7 ±3.6	101.3 ±3.6	47.4 ±2.1	0.7435 ±0.018	0.852 ±0.015	0.7506 ±0.027	0.8294 ±0.013
33	2	138.2 ±3.8	61.8 ±3.8	27.5 ±2.9	24.4 ±3.1	110.6 ±3.1	37.5 ±2.9	0.7405 ±0.023	0.8009 ±0.019	0.8195 ±0.023	0.8045 ±0.017
34	2	120.3 ±3.6	79.7 ±3.6	19.6 ±2.5	34.3 ±2.9	100.7 ±2.9	45.4 ±2.5	0.7303 ±0.02	0.8373 ±0.018	0.7457 ±0.021	0.8171 ±0.017
35	2	122.9 ±3.3	77.1 ±3.3	18.4 ±2.4	30.4 ±2.5	104.6 ±2.5	46.6 ±2.4	0.756 ±0.019	0.8508 ±0.018	0.7746 ±0.019	0.8343 ±0.015
36	2	112.5 ±3.6	87.5 ±3.6	15.7 ±1.6	38.2 ±3.3	96.8 ±3.3	49.3 ±1.6	0.7305 ±0.018	0.8605 ±0.013	0.717 ±0.024	0.8272 ±0.013
37	2	110.1 ±3.2	89.9 ±3.2	14.1 ±2.0	39.0 ±2.6	96.0 ±2.6	50.9 ±2.0	0.7345 ±0.016	0.8721 ±0.015	0.7111 ±0.019	0.8342 ±0.013

Continued on next page

Table A.7 – Continued from previous page

ID-I	ID-S	PP	PN	FP	FN	TP	TN	accuracy	precision	recall	$F_{0.5}$
38	2	133.8 ±3.9	66.2 ±3.9	25.9 ±2.2	27.0 ±3.0	108.0 ±3.0	39.1 ±2.2	0.7355 ±0.018	0.8069 ±0.014	0.7998 ±0.022	0.8053 ±0.013
39	2	110.9 ±4.1	89.1 ±4.1	15.2 ±2.1	39.3 ±3.4	95.7 ±3.4	49.8 ±2.1	0.7275 ±0.02	0.8631 ±0.017	0.7089 ±0.025	0.8269 ±0.015
40	2	103.9 ±4.4	96.1 ±4.4	11.6 ±2.0	42.7 ±3.9	92.3 ±3.9	53.4 ±2.0	0.7287 ±0.021	0.8886 ±0.017	0.684 ±0.029	0.8381 ±0.016

Table A.6: Results for simulated parameter combinations listed in table A.4.

ID-S	ID-I	PP	PN	FP	FN	TP	TN	accuracy	precision	recall	$F_{0.5}$
0	0	83.07 ±2.86	116.93 ±2.86	11.33 ±1.73	63.27 ±2.69	71.73 ±2.69	53.67 ±1.73	0.63 ±0.02	0.86 ±0.02	0.53 ±0.02	0.77 ±0.02
1	0	103.17 ±3.36	96.83 ±3.36	13.03 ±1.63	44.87 ±3.08	90.13 ±3.08	51.97 ±1.63	0.71 ±0.02	0.87 ±0.01	0.67 ±0.02	0.82 ±0.01
2	0	107.13 ±3.17	92.87 ±3.17	13.57 ±1.43	41.43 ±2.24	93.57 ±2.24	51.43 ±1.43	0.72 ±0.01	0.87 ±0.01	0.69 ±0.02	0.83 ±0.01
3	0	106.87 ±6.75	93.13 ±6.75	12.67 ±2.02	40.8 ±5.32	94.2 ±5.32	52.33 ±2.02	0.73 ±0.02	0.88 ±0.01	0.7 ±0.04	0.84 ±0.01
4	0	114.13 ±3.16	85.87 ±3.16	15.33 ±1.79	36.2 ±3.08	98.8 ±3.08	49.67 ±1.79	0.74 ±0.02	0.87 ±0.01	0.73 ±0.02	0.84 ±0.01
5	0	99.97 ±4.33	100.03 ±4.33	14.6 ±1.77	49.63 ±3.92	85.37 ±3.92	50.4 ±1.77	0.68 ±0.02	0.85 ±0.02	0.63 ±0.03	0.8 ±0.02
6	0	84.67 ±1.18	115.33 ±1.18	9.7 ±0.75	60.03 ±1.52	74.97 ±1.52	55.3 ±0.75	0.65 ±0.01	0.89 ±0.01	0.56 ±0.01	0.79 ±0.01
7	0	104.67 ±3.12	95.33 ±3.12	12.33 ±1.95	42.67 ±2.04	92.33 ±2.04	52.67 ±1.95	0.72 ±0.01	0.88 ±0.02	0.68 ±0.02	0.83 ±0.01
8	0	112.43 ±3.09	87.57 ±3.09	20.2 ±2.22	42.77 ±1.57	92.23 ±1.57	44.8 ±2.22	0.69 ±0.01	0.82 ±0.02	0.68 ±0.01	0.79 ±0.01
9	0	109.73 ±3.33	90.27 ±3.33	15.2 ±1.13	40.47 ±3.13	94.53 ±3.13	49.8 ±1.13	0.72 ±0.02	0.86 ±0.01	0.7 ±0.02	0.82 ±0.01
10	0	104.77 ±3.19	95.23 ±3.19	12.67 ±2.15	42.9 ±2.72	92.1 ±2.72	52.33 ±2.15	0.72 ±0.02	0.88 ±0.02	0.68 ±0.02	0.83 ±0.02
11	0	106.47 ±3.65	93.53 ±3.65	12.97 ±1.73	41.5 ±3.03	93.5 ±3.03	52.03 ±1.73	0.73 ±0.02	0.88 ±0.01	0.69 ±0.02	0.83 ±0.01
12	0	107.23 ±3.89	92.77 ±3.89	14.03 ±1.65	41.8 ±3.55	93.2 ±3.55	50.97 ±1.65	0.72 ±0.02	0.87 ±0.01	0.69 ±0.03	0.83 ±0.01
13	0	106.03 ±4.24	93.97 ±4.24	13.1 ±1.75	42.07 ±3.26	92.93 ±3.26	51.9 ±1.75	0.72 ±0.02	0.88 ±0.01	0.69 ±0.02	0.83 ±0.01
14	0	15.53 ±2.36	184.47 ±2.36	0.1 ±0.31	119.57 ±2.43	15.43 ±2.43	64.9 ±0.31	0.4 ±0.01	0.99 ±0.02	0.11 ±0.02	0.39 ±0.04
15	0	107.33 ±3.84	92.67 ±3.84	13.33 ±2.55	41.0 ±3.49	94.0 ±3.49	51.67 ±2.55	0.73 ±0.02	0.88 ±0.02	0.7 ±0.03	0.83 ±0.02
16	0	106.17 ±3.59	93.83 ±3.59	12.67 ±2.14	41.5 ±3.44	93.5 ±3.44	52.33 ±2.14	0.73 ±0.02	0.88 ±0.02	0.69 ±0.03	0.84 ±0.02
17	0	107.2 ±3.96	92.8 ±3.96	13.83 ±2.15	41.63 ±3.15	93.37 ±3.15	51.17 ±2.15	0.72 ±0.02	0.87 ±0.02	0.69 ±0.02	0.83 ±0.02
18	0	105.47 ±4.08	94.53 ±4.08	12.97 ±2.11	42.5 ±3.4	92.5 ±3.4	52.03 ±2.11	0.72 ±0.02	0.88 ±0.02	0.69 ±0.03	0.83 ±0.02
19	0	108.33 ±1.86	91.67 ±1.86	13.67 ±1.06	40.33 ±1.45	94.67 ±1.45	51.33 ±1.06	0.73 ±0.01	0.87 ±0.01	0.7 ±0.01	0.83 ±0.01
20	0	107.77 ±3.4	92.23 ±3.4	13.63 ±1.43	40.87 ±2.53	94.13 ±2.53	51.37 ±1.43	0.73 ±0.01	0.87 ±0.01	0.7 ±0.02	0.83 ±0.01

A.5 Algorithms

Algorithm 3 Find *Worst* Pre-Transfer State (Global)

Input: Full result set Y_{oV}, see Equation 6.28.
Output: WorstStateGlobal

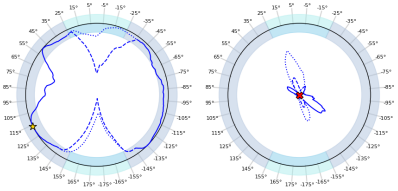
```
WorstStateGlobal ← (−1, −1)          ▷ Initialize WorstStateGlobal with a placeholder
MinValueGlobal ← ∞
MinHalfWindowSize ← 0
for all Ȳ^MA ∈ Y_oV do
    LocalResult ← Call Algorithm 4 with Ȳ^MA as input .
    WorstStateLocal ← LocalResult[1]
    AppliedHalfWindowSize ← LocalResult[2]
    if WorstStateLocal[1] < MinValueGlobal then
        WorstStateGlobal ← WorstStateLocal
        MinHalfWindowSize ← AppliedHalfWindowSize
    else if WorstStateLocal[1] = WorstStateGlobal then
        if AppliedHalfWindowSize < MinHalfWindowSize then
            WorstStateGlobal ← WorstStateLocal
            MinHalfWindowSize ← AppliedHalfWindowSize
        end if
    end if
end for
```

Algorithm 4 Find *Worst* Pre-Transfer State (Local)

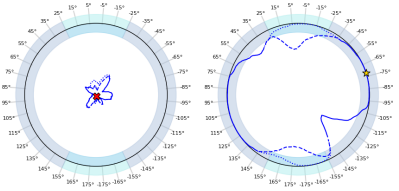
Input: Single result set $\overline{Y}^{MA} \in Y_{o^V}$, see Equation 6.28.
Output: WorstStateLocal, AppliedHalfWindowSize

```
MinValue ← ∞
MinEntries ← ∅
for each (ȳ^MA, θ) ∈ Ȳ^MA do
    if ȳ^MA < MinValue then
        MinValue ← ȳ^MA
        MinEntries ← {(ȳ^MA, θ)}
    else if ȳ^MA = MinValue then
        Add (ȳ^MA, θ) to MinEntries
    end if
end for
HalfWindowSize ← 1
while |MinEntries| > 1 do
    MinFilteredValue ← ∞
    FilteredEntries ← ∅
    for each (ȳ^MA, θ) ∈ MinEntries do
        Apply moving average filter on initial set Ȳ^MA
        FilteredValue ← 1/(HalfWindowSize∗2+1) Σ_{j=−HalfWindowSize}^{HalfWindowSize} ȳ^MA_{i+j}
        if FilteredValue < MinFilteredValue then
            MinFilteredValue ← FilteredValue
            FilteredEntries ← {(ȳ^MA, θ)}
        else if FilteredValue = MinFilteredValue then
            Add (ȳ^MA, θ) to FilteredEntries
        end if
    end for
    MinEntries ← FilteredEntries
    HalfWindowSize ← HalfWindowSize + 1
end while
assert |MinEntries| = 1
WorstStateLocal ← MinEntries[1]
AppliedHalfWindowSize ← HalfWindowSize[1]
```

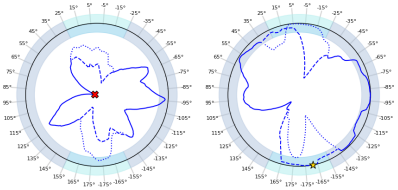
A.6 Full state experiments result plot



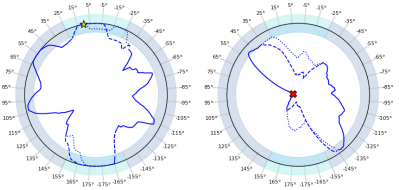
(a) o_1^V



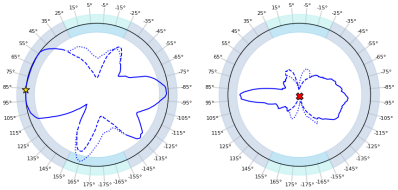
(b) o_2^V



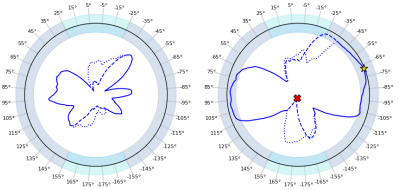
(c) o_3^V



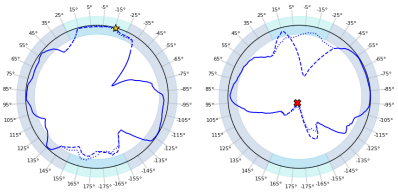
(d) o_4^V



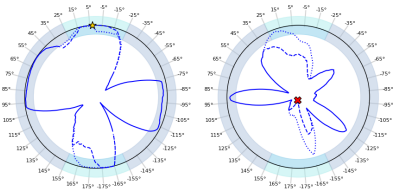
(e) o_5^V



(f) o_6^V

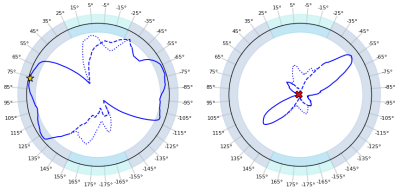


(g) o_7^V

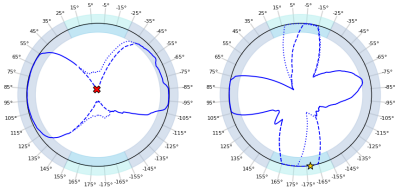


(h) o_8^V

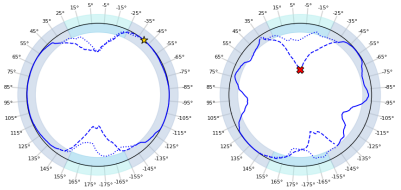
Figure A.1: Full state experiment results Y_{o^V} for each $o^V \in \mathcal{O}^V$.



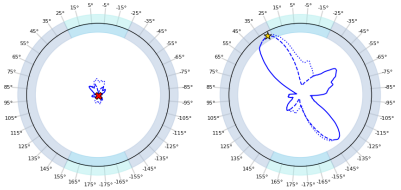
(i) o_9^V



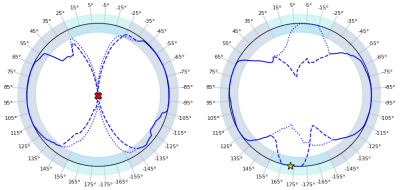
(j) o_{10}^V



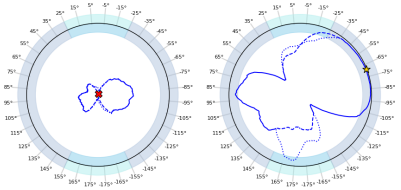
(k) o_{11}^V



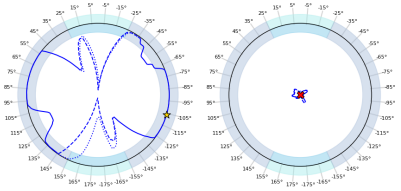
(l) o_{12}^V



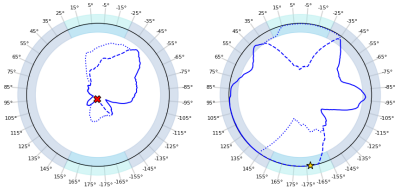
(m) o_{13}^V



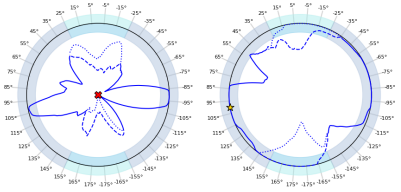
(n) o_{14}^V



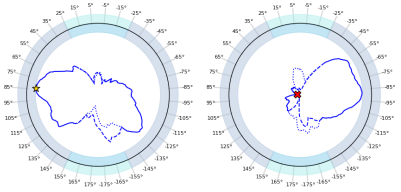
(o) o_{15}^V



(p) o_{16}^V

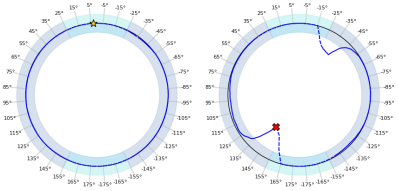


(q) o_{17}^V

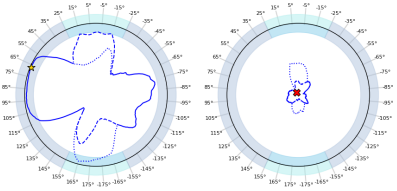


(r) o_{18}^V

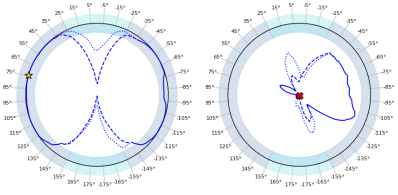
Full state experiment results Y_{o^V} for each $o^V \in \mathcal{O}^V$.



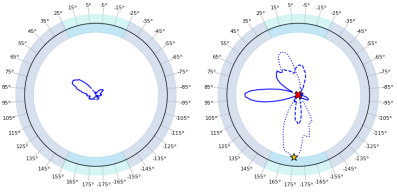
(s) o_{19}^V



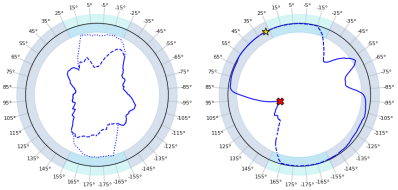
(t) o_{20}^V



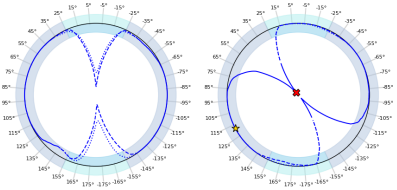
(u) o_{21}^V



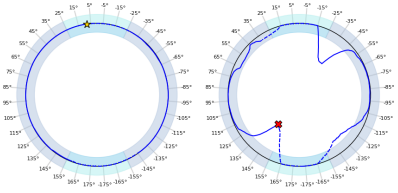
(v) o_{22}^V



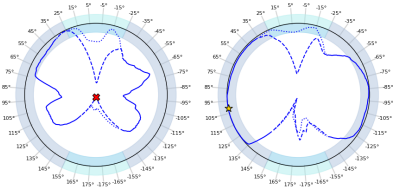
(w) o_{23}^V



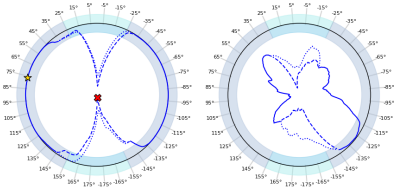
(x) o_{24}^V



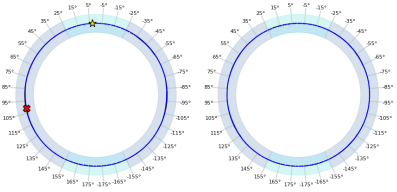
(y) o_{25}^V



(z) o_{26}^V

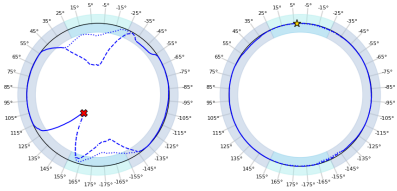


(aa) o_{27}^V

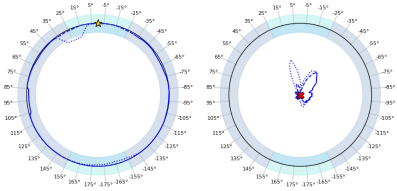


(ab) o_{28}^V

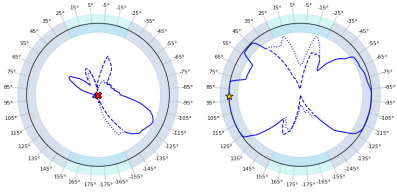
Full state experiment results Y_{o^V} for each $o^V \in \mathcal{O}^V$.



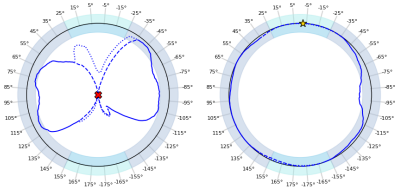
(ac) o_{29}^{V}



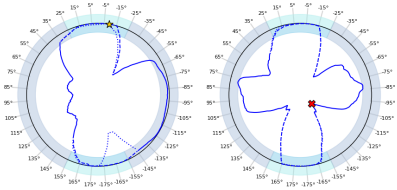
(ad) o_{30}^{V}



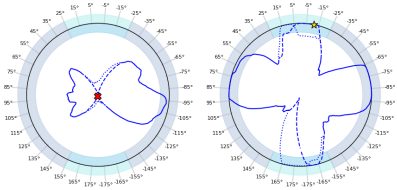
(ae) o_{31}^{V}



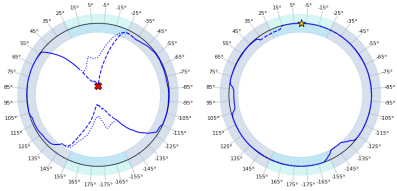
(af) o_{32}^{V}



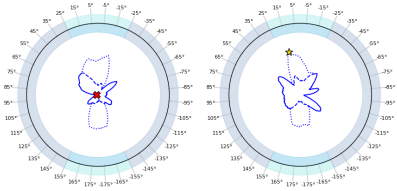
(ag) o_{33}^{V}



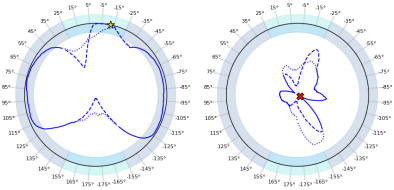
(ah) o_{34}^{V}



(ai) o_{35}^{V}



(aj) o_{36}^{V}



(ak) o_{37}^{V}

Full state experiment results Y_{o^V} for each $o^V \in \mathcal{O}^V$.

List of Acronyms and Symbols

General Abbreviations

DT	Digital Twin	7
AGV	Automated Guided Vehicle	9
AMR	Autonomous Mobile Robot	9
DES	Discrete Event Simulation	9
UML	Unified Modeling Language	14
CPS	Cyber-Physical System	14
RAMI4.0	Reference Architectural Model Industry 4.0	19
AAS	Asset Administration Shell	19
HAL	Hardware Abstraction Layer	21
KG	Knowledge Graph	26
SoS	System of Systems	30
RFID	Radio Frequency Identification	34
KIT	Karlsruhe Institute of Technology	37
OPC	Open Platform Communications	40
OPC UA	OPC Unified Architecture	40
RGB-D	Combined colour and a depth sensor	41
VR	Virtual Reality	71
STL	File format to a stereolithography model	46

Digital Twin Architecture

$\mathcal{A}^{\mathcal{P}}$	Set of physical assets	21
$\mathcal{A}^{\mathcal{L}}$	Set of logical assets	25
$\mathcal{A}^{\mathcal{V}}$	Set of virtual models	25

Formal Object Transfer System

OTS	Object Transfer System	30
o	Object o which can be transferred	30
$\mathcal{O}$	Set of objects to be transferred	30
a_{pre}	Asset which possesses o when the OTS is formed	30
a_{post}	Asset which is supposed to possess o after a successful transfer	30
x_i	Components of the state definition of o	30
a	Specific asset a	31
s	State vector	31
$\mathcal{S}_o$	Set of all possible states of o during the lifetime of an OTS	31
$\mathcal{S}_o^{a_{\text{pre}}}$	Set of states where o is possessed by a_{pre}	31
$\mathcal{S}_o^{a_{\text{post}}}$	Set of states where o is possessed by a_{post}	31
s	Object state	31
$\mathcal{S}_o^{\text{pre}}$	Set of pre-transfer states	31
P^{pre}	Predicate that evaluates pre-transfer states	31
P^{+}	Predicate that evaluates success states	32
P^{-}	Predicate that evaluates failure states	32
$\mathcal{S}_o^{+}$	Set of success states of o	32
$\mathcal{S}_o^{-}$	Set of failure states of o	32
$\mathcal{S}_o^{'}$	Set of non terminating states of o	32
t	Time parameter	32

t_0	Time at the start of the object transfer process 32
t_{end}	Time at the end of the object transfer process 32
$s(t)$	Function which describes the state over time. 32
u	State transition function 32
$\mathcal{U}$	Set of factors that influence the object transfer process 32
$s[t_i]$	Sequence of states at discrete times t_i 33
$p^+(s[t_0])$	Estimated probability of a successful object transfer process given a pre-transfer state 33

Implemented Object Transfer System

AgiProbot	Agile Production System using Mobile, Learning Robots with Multisensor Technology for Uncertain Production Specifications 37
$\mathcal{TU}^P$	Real-world transfer unit 40
$\mathcal{TU}^P$	Set of real-world transfer units 40
TU_B^P	Specific transfer unit mounted on Station B 40
TU_C^P	Specific transfer unit mounted on Station C 40
TU_D^P	Specific transfer unit mounted on Station D 40
$\mathcal{TM}^P$	Set of real-world transport modules 41
TM_1^P	Implemented real-world transport module 41
$\mathcal{O}^P$	Set of real-world starter motor cores 42
o^P	Specific real-word starter motor core 42
$\mathcal{TM}^V$	Set of virtual transport modules 45
TM^V	A virtual transport module 45
$\mathcal{TU}^V$	Set of virtual transfer units 45
TU^V	A virtual transfer unit 45
$\mathcal{O}^V$	Set of virtual starter motor cores 46

o^V	Virtual starter motor core i 46
F_T	Fixed frame defined at the front tip of TM_1^P 52
F_S	Fixed frame defined at the mounting point of the servo module 52
F_A	Rotatable frame defined in the axis of rotation of the servomotor52
F_C	Fixed camera frame with z-axis as optical axis 52
F_{BB}	Frame at the centre of an Axis Aligned Bounding Box 52
F_O	Object specific frame for the starter motor set 52
AABB	Axis aligned Bounding Box 53
x	Translation in x-direction of F_{BB} relative to F_T 53
y	Translation in y-direction of F_{BB} relative to F_T 53
z	Translation in z-direction of F_{BB} relative to F_T 53
s_x	Half extend in x-direction of the AABB 53
s_y	Half extend in y-direction of the AABB 53
s_z	Half extend in z-direction of the AABB 53
θ	Yaw angle of F_O relative to F_T 54
f	Face on which the object is placed on 54
c	Conveyor position of the object 54
t_{max}	Timeout time 56
RANSAC	Radom Sample Consensus algorithm 58
$\mathcal{O}_{detected}$	Set of detected objects 58
$\mathcal{O}_{tracked}$	Set of tracked objects 58
α	Servo angle 60
${}_C\vec{\boldsymbol{d}}_{CBB}$	Homogeneous vector from F_C to F_{BB}, expressed in F_C 60
d_z	z-component of ${}_C\vec{\boldsymbol{d}}_{CBB}$ 60
${}_{BB}\vec{\boldsymbol{d}}_{CBB}$	Homogeneous vector from F_C to F_{BB}, expressed in F_{BB} 60
$\boldsymbol{T}_{i,j}$	Transformation from frame i to frame j 60

Application Study

$\hat{p}^V$	Estimated probability of success 68
$s^{\text{pre}}_{o,\text{best}}$	Best pre-transfer state for object o^P 68
$s^{\text{pre}}_{o,\text{worst}}$	Worst pre-transfer state for object o^P 68
${}^E\mathcal{S}^{\text{pre}}_o$	Set of pre-transfer states used in the experiments of the application study 69
p_{left}	Fixed position in the centre of the left conveyor belt. 70
p_{middle}	Fixed position in the centre of the both conveyor belt. 70
p_{right}	Fixed position in the centre of the right conveyor belt. 70
${}^E\Theta$	Set of discretised yaw angles θ_i 70
${}^E\Theta_{\text{single}}$	Set of discretised yaw angles used at p_{left} and p_{right} 70
${}^E\Theta_{\text{both}}$	Set of discretised yaw angles used at p_{middle} 70
${}^E\mathcal{S}^{\text{pre}}_{o,\text{left}}$	Set of pre-transfer states considered at p_{left} 71
${}^E\mathcal{S}^{\text{pre}}_{o,\text{both}}$	Set of pre-transfer states considered at p_{middle} 71
${}^E\mathcal{S}^{\text{pre}}_{o,\text{right}}$	Set of pre-transfer states considered at p_{right} 71
ChArUco	Fiducial markers (ArUco) positioned on a chessboard 71
${}_e\tilde{\xi}$	Measured docking error 71
${}_e\tilde{\xi}_x$	Measured docking error, x-component 71
${}_e\tilde{\xi}_y$	Measured docking error, y-component 71
${}_e\tilde{\xi}_\theta$	Measured docking error, yaw angle 71
F_{C_T}	Fixed frame in the camera module 72
F_B	Frame defined at the corner of the detected 5x5 ChArUco board 72
F_{B_G}	Fixed frame defined at the goal pose of the ChArUco board . . . 72
F_D	Fixed frame centered on the transport module. 72
F_{D_G}	Frame at the goal docking pose. 72
E^P	Real world experiment 74

y_1^P	Result of the real-world transfer process from a transport module to a transfer unit 74
y_2^P	Result of the real-world transfer process from a transfer unit to a transport module 74
y^P	Combined result of a real-world object transfer experiment 74
E^V	Virtual experiment i 75
y^V	Combined result of a virtual object transfer experiment 76
y_1^V	Result of the virtual transfer process from a transport module to a transfer unit 76
y_2^V	Result of the virtual transfer process from a transfer unit to a transport module 76
${}_e\xi$	Docking error 75
${}_e\xi_x$	Docking error, x-component 83
${}_e\xi_y$	Docking error, y-component 83
${}_e\xi_\theta$	Docking error, z-component 83
CDF	Cumulative Distribution Function 83
Φ_ρ	CDF of a multivariate normal distribution 83
$\boldsymbol{\rho}$	Correlation matrix 83
ϕ^{-1}	Inverse of the univariate standard normal CDF 83
$F_i(X_i)$	CDF of the random variable X_i 83
μ_k	Dynamic friction coefficient 89
d_{gap}	Initial gap distance 89
PP	Predicted positives 91
PN	Predicted negatives 91
FP	False positives 91
FN	False negatives 91
TP	True positives 91

TP	True negatives 91
${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{front,left}}$	Set of pre-transfer states on the front face including the left conveyor belt 93
${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{front,right}}$	Set of pre-transfer states on the front face including the right conveyor belt 93
${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{back,left}}$	Set of pre-transfer states on the back face including the left conveyor belt 93
${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{back,right}}$	Set of pre-transfer states on the back face including the right conveyor belt 94
$Y_{o^V,r,\text{front,left}}$	Series of results for run r of o^V for pre-transfer states ${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{front,left}}$ 94
$\overline{y}^V$	Mean virtual transfer results 94
$\overline{Y}_{o^V,\text{front,left}}$	Series of mean results for o^V in pre-transfer states ${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{front,left}}$. 94
$\overline{y}^{MA}_i$	Moving average result i 95
$\overline{Y}^{MA}_{o^V,\text{front,left}}$	Series of moving average results of o^V for pre-transfer states ${}^{E}\mathcal{S}^{\text{pre}}_{o,\text{front,left}}$ 95
Y_{o^V}	Combined series of results for o^V 96

List of Figures

1.1 Supply chain of a *Circular Product Service System.* 2
1.2 The remanufacturing process chain. 2
2.1 *Digital Twin* concept. 8
2.2 PRISMA chart of the structured literature review. 12
3.1 Different types of physical assets and their virtual representations. . 23
3.2 Detailed view of a type **E** asset and its virtual representation. 24
3.3 Logical Assets. 26
3.4 The proposed *Digital Twin* architecture. 27
4.1 Euler diagram to visualise the relationship of different object state subsets. 31
4.2 Schematic representation of an exemplary object transfer process. . . 33
4.3 Exemplary object transfer systems scenarios. 34
5.1 Main research areas of the AgiProbot project. 38
5.2 The *AgiProbot* production system, see (Wurster et al. 2021, Klein et al. 2021). 39
5.3 A transfer unit, a mixed type **D/E** physical asset. 42
5.4 A transport module, a type **E** physical asset. 43
5.5 Set of unique starter motor cores $\mathcal{O}^P$. 44
5.6 Virtual models in an NVIDIA Isaac simulation environment. 45
5.7 Set of virtual starter motor cores $\mathcal{O}^V$. 47
5.8 Implemented CPS, part of the *AgiProbot* demonstration system. . . 49
5.9 The transport module (TM_1^P) after a successful docking. 51
5.10 Navigation modes of the implemented transport module TM_1^P. . . . 52
5.11 Coordinate frames as listed in Table 5.1. 53
5.12 Examples of different pre-transfer states. 55

5.13 Examples of success and failure states. 57
5.14 Implemented detection pipeline. 59
5.15 Visual servoing results. 61
5.16 Schematic view on different transfer unit configurations. 62
5.17 Multi object detection challenges. 63
5.18 Gap area between two aligned conveyor belts, see (Klein and Furmans 2025) 64
6.1 Pre-transfer state examples $s^{\text{pre}} \in {}^{E}\mathcal{S}_{o}^{\text{pre}}$. 72
6.2 Camera-based set up to evaluate the docking pose error. 73
6.3 Procedure of a real-world object transfer experiment. 75
6.4 Procedure of a virtual object transfer experiment. 76
6.5 Flowchart for generating the *Calibration Experiments*. 78
6.6 Exemplary labeled result plot for o_1^P. 79
6.7 Real world experiment results for four different starter motor core variants. 80
6.8 Image series of selected *Calibration Experiment* result states. 81
6.9 Histograms of the measured docking errors. 82
6.10 Correlation matrix of the measured docking errors. 82
6.11 Synthetically generated docking pose errors. 84
6.12 Results of *Virtual Calibration Experiments* for different simulation parameters. 88
6.13 Visualisation of the *Inital Gap Distance* parameter. 89
6.14 Experiment results for single instance parameter variations. 90
6.15 Results for different instance parameter combinations. 90
6.16 Sim-To-Real result comparison for a single virtual calibration. . . . 92
6.17 Binary results for a single *Full State Run*: $Y_{o_1^V,1,\text{front,left}}$. 94
6.18 Mean transfer results $\overline{Y}_{o_1^V,\text{front,left}}$. 95
6.19 Results with applied moving average filter $\overline{Y}^{MA}_{o_1^V,\text{front,left}}$. 95
6.20 Exemplary result plot for the *Full State Experiments* of o^V. 97
6.21 Examples for $s^{\text{pre}}_{o,\text{best}}$ and $s^{\text{pre}}_{o,\text{worst}}$ determined for o_1^V. 98
6.22 95% confidence intervals including the estimated probability of success $\hat{p}^V$ for $s^{\text{pre}}_{o,\text{best}}$ for each o^V. 101

6.23 95% confidence intervals including the estimated probability of success $\hat{p}^V$ for $s^{\text{pre}}_{o,\text{worst}}$ for each o^V. 102
6.24 Confidence interval comparison between virtual and physical results for $s^{\text{pre}}_{o,\text{best}}$. 103
6.25 Confidence interval comparison between virtual and physical results for $s^{\text{pre}}_{o,\text{worst}}$. 103
6.26 Examples for an inaccurate mass distribution. 104
6.27 Examples for inaccurate virtual object models. 105
6.28 Examples for an unstable physics simulation. 105
A.1 Full state experiment results Y_{o^V} for each $o^V \in \mathcal{O}^V$. 128

List of Tables

2.1 Categorised research publications on digital twins in the context of remanufacturing and disassembly automation. 13
3.1 Comparison of different asset types. 22
5.1 Overview of the relevant coordinate frames in a formed OTS. 52
6.1 Overview of the relevant coordinate frames used to measure the docking error of TM_1^P. 73
6.2 Experiment description: *Calibration Experiments*. 77
6.3 Experiment description: *Virtual Calibration Experiments*. 85
6.4 Metrics of a single series of *Virtual Calibration Runs* with the selected parameter sets (ID-S 2, ID-I 23). 91
6.5 Experiment description: *Full State Experiments*. 93
6.6 Experiment description: *Validation Experiments*. 102
A.1 Quantitative calibration experiment results 113
A.2 Confusion Matrix. 119
A.3 Constant simulation parameters. 119
A.4 Evaluated simulation parameter combinations (ID-S). 120
A.5 Instance parameter combinations (ID-I). 121
A.7 Evaluation of instance parameter combinations, see Table A.5. . . . 122
A.6 Results for simulated parameter combinations listed in table A.4. . . 125

List of Publications

Behrendt, S., M. Wurster, M. Strljic, J.-F. Klein, M. C. May and G. Lanza. 2023. Interoperable Architecture For Logical Reconfigurations Of Modular Production Systems. In *Proceedings of the 5th Conference on Production Systems and Logistics (CPSL-2). Ed.: D. Herberger,* p. 622–633. Proceedings of the Conference on Production Systems and Logistics. Offenburg, Germany: publish-Ing.

Enke, C., J.-F. Klein, M. Sperling, B. Zhou and K. Furmans. 2022. Development of an Experimental Environment to Study the Challenges in Cyber-Physical Intralogistics Systems. In *18. Fachkolloquium der Wissenschaftlichen Gesellschaft für Technische Logistik e. V.* Logistics Journal: Proceedings.

Fleischer, J., F. Zanger, V. Schulze, G. Neumann, N. Stricker, K. Furmans, J. Pfrommer, G. Lanza, M. Hansjosten, P. Fischmann, J. Dvorak, J.-F. Klein, F. Rauscher, A. Ebner, M. C. May and P. Gönnheimer. 2024. Self-learning and autonomously adapting manufacturing equipment for the circular factory. *at - Automatisierungstechnik* 72 (9): p. 861–874.

Hofmann, C., S. Staab, M. Selzer, G. Neumann, K. Furmans, M. Heizmann, J. Beyerer, G. Lanza, J. Pfrommer, T. Düser and J.-F. Klein. 2024. The role of an ontology-based knowledge backbone in a circular factory. *at - Automatisierungstechnik* 72 (9): p. 875–883.

Klein, J.-F. and K. Furmans. 2023. Digital twin architecture and sim-to-real gap analysis of a material transfer system in a remanufacturing environment. In *Procedia CIRP,* edited by D. O. Mpofu K. Sacks N., 120:p. 368–373. Elsevier B.V.

Klein, J.-F., C. Enke and M. Ries. 2022. PropS: Towards Proprioception in Cyber-Physical Production Systems by Means of Collaborative Localization. In *2022 IEEE International Systems Conference (SysCon),* p. 1–6. IEEE.

Klein, J.-F. and K. Furmans. 2025. A Study on the Predictive Capabilities of Digital Twins for Object Transfers in Fluid Automated Remanufacturing. *Unpublished working paper.*

Klein, J.-F., M. Wurster, N. Stricker, G. Lanza and K. Furmans. 2021. Towards Ontology-based Autonomous Intralogistics for Agile Remanufacturing Production Systems. In *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA),* p. 1–7. IEEE.

Lanza, G., T. Asfour, J. Beyerer, B. Deml, J. Fleischer, M. Heizmann, K. Furmans, C. Hofmann, A. Cebulla, C. Dreher, J.-P. Kaiser, J.-F. Klein, F. Leven, S. Mangold, N. Mitschke, N. Stricker, J. Pfrommer, C. Wu, M. Wurster and M. Zaremski. 2022. Agiles Produktionssystem mittels lernender Roboter bei ungewissen Produktzuständen am Beispiel der Anlasser-Demontage. *at - Automatisierungstechnik* 70 (6): p. 504–516.

Pfrommer, J., J.-F. Klein, M. Wurster, S. Rapp, P. Grauberger, G. Lanza, A. Albers, S. Matthiesen and J. Beyerer. 2022. An ontology for remanufacturing systems. *at - Automatisierungstechnik* 70 (6): p. 534–541.

Sprenger, K., J.-F. Klein, M. Wurster, N. Stricker and G. Lanza. 2021. Industrie 4.0 im Remanufacturing. *Industrie 4.0 Management* 2021 (4): p. 37–40.

Wu, C., K. Zhou, J.-P. Kaiser, N. Mitschke, J.-F. Klein, J. Pfrommer, J. Beyerer, G. Lanza, M. Heizmann and K. Furmans. 2022. MotorFactory: A Blender Add-on for Large Dataset Generation of Small Electric Motors. *Procedia CIRP* 106:p. 138–143.

Wurster, M., J.-F. Klein, J.-P. Kaiser, S. Mangold, K. Furmans, M. Heizmann, J. Fleischer and G. Lanza. 2022. Integrierte Steuerungsarchitektur für ein agiles Demontagesystem mit autonomer Produktbefundung. *at - Automatisierungstechnik* 70 (6): p. 542–556.

Zhou, B., J.-F. Klein, B. Wang and M. Hillemann. 2023. Semantic Mapping and Autonomous Navigation for Agile Production System. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE),* p. 1–6. IEEE.

Zhou, B., A. Wang, J.-F. Klein and F. Kai. 2021. Object Detection and Mapping with Bounding Box Constraints. In *2021 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI),* p. 1–6. IEEE.

Bibliography

Ashtari Talkhestani, B., T. Jung, B. Lindemann, N. Sahlab, N. Jazdi, W. Schloegl and M. Weyrich. 2019. An architecture of an Intelligent Digital Twin in a Cyber-Physical Production System. *at - Automatisierungstechnik* 67 (9): p. 762–782.

Assuad, C., T. Leirmo and K. Martinsen. 2022. Proposed framework for flexible de- and remanufacturing systems using cyber-physical systems, additive manufacturing, and digital twins. In *Procedia CIRP,* edited by D. D. Teti R., 112:p. 226–231. Elsevier B.V.

Bader, S. R. and M. Maleshkova. 2019. The Semantic Asset Administration Shell. In *Semantic Systems. The Power of AI and Knowledge Graphs,* edited by M. Acosta, P. Cudré-Mauroux, M. Maleshkova, T. Pellegrini, H. Sack and Y. Sure-Vetter, 11702:p. 159–174. Lecture Notes in Computer Science. Springer International Publishing.

Bras, A. and P. Neto. 2018. Unsupervised Feature Extraction from RGB-D Data for Object Classification: a Case Study on the YCB Object and Model Set. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society,* p. 3673–3678. IEEE.

Cebulla, A., T. Asfour and T. Kröger. 2023. Speeding Up Assembly Sequence Planning Through Learning Removability Probabilities. In *2023 IEEE International Conference on Robotics and Automation (ICRA),* p. 12388–12394. IEEE.

Chen, Z. and L. Huang. 2021. Digital twins for information-sharing in remanufacturing supply chain: A review. *Energy* 220.

Cho, S., G. May and D. Kiritsis. 2019. A Semantic-driven Approach for Industry 4.0. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS),* p. 347–354.

Clopper, C. J. and E. S. Pearson. 1934. The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial. *Biometrika* 26 (4): p. 404–413.

DIN SPEC 91472:2023-06. 2023. Remanufacturing (Reman) - Qualitätsklassifizierung für zirkuläre Prozesse. Berlin: Beuth Verlag GmbH.

Dreher, C. R. and T. Asfour. 2022. Learning Temporal Task Models from Human Bimanual Demonstrations. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* p. 7664–7671. IEEE.

Durante, F. and C. Sempi. 2016. Principles of copula theory. Boca Raton, FL: CRC Press.

Elfaham, H. and U. Epple. 2020. Meta models for intralogistics. *at - Automatisierungstechnik* 68 (3): p. 208–221.

Elsner, J., A. Gabriel, T. Ackermann, J. Körkemeyer and R. Schmitt. 2023. Digital Twin-based Life Cycle Assessment [Auf Digitalen Zwillingen basiertes Life Cycle Assessment Integration des Digitalen Zwillings für die dynamische Ökobilanzierung im variantenreichen Remanufacturing von Batterien]. *ZWF Zeitschrift fuer Wirtschaftlichen Fabrikbetrieb* 118 (12): p. 883–887.

Fagerland, M. W., S. Lydersen and P. Laake. 2015. Recommended confidence intervals for two independent binomial proportions. *Statistical Methods in Medical Research* 24 (2): p. 224–254.

Ferko, E., A. Bucaioni, P. Pelliccione and M. Behnam. 2023. Standardisation in Digital Twin Architectures in Manufacturing. In *2023 IEEE 20th International Conference on Software Architecture (ICSA),* p. 70–81.

Frenz, W. 2022. Bundes-Klimaschutzgesetz (KSG). In *Klimaschutzrecht,* edited by W. Frenz, p. 525–841. Berlin: Erich Schmidt Verlag GmbH & Co. KG.

Furmans, K. and K. R. Gue. 2018. A Framework for Modeling Material Handling with Decentralized Control. In *15th IMHRC Proceedings*. Savannah, Georgia, USA.

Ghorbani, H. and F. Khameneifar. 2022. Construction of damage-free digital twin of damaged aero-engine blades for repair volume generation in remanufacturing. *Robotics and Computer-Integrated Manufacturing* 77.

Glaessgen, E. and D. Stargel. 2012. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. American Institute of Aeronautics / Astronautics.

Glatt, M., D. Kull, B. Ravani and J. C. Aurich. 2019. Validation of a physics engine for the simulation of material flows in cyber-physical production systems. *Procedia CIRP,* 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019, 81:p. 494–499.

Glatt, M., C. Sinnwell, L. Yi, S. Donohoe, B. Ravani and J. C. Aurich. 2021. Modeling and implementation of a digital twin of material flows based on physics simulation. *Journal of Manufacturing Systems* 58:p. 231–245.

Grieves, M. W. 2005. Product lifecycle management: the new paradigm for enterprises. *International Journal of Product Development* 2 (1/2): p. 71–84.

Gu, Q. and G. Tagaras. 2014. Optimal collection and remanufacturing decisions in reverse supply chains with collector's imperfect sorting. *International Journal of Production Research* 52 (17): p. 5155–5170.

Guide, V. R. 2000. Production planning and control for remanufacturing: industry practice and research needs. *Journal of Operations Management* 18 (4): p. 467–483.

Guo, H.-F. and Y. Han. 2022. Research on Closed-Loop Supply Chain Based on Digital Twin. In *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence, AHPCAI 2022,* p. 270–273. IEEE.

Hauge, J. B., M. Zafarzadeh, Y. Jeong, Y. Li, W. A. Khilji and M. Wiktorsson. 2020. Employing digital twins within production logistics. In *2020 IEEE International Conference on Engineering, Technology and Innovation (ICE/ITMC),* p. 1–8. IEEE.

Hogan, A., E. Blomqvist, M. Cochez, C. D'amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, A.-C. N. Ngomo, A. Polleres, S. M. Rashid, A. Rula, L. Schmelzeisen, J. Sequeda, S. Staab and A. Zimmermann. 2021. Knowledge Graphs. *ACM Computing Surveys* 54 (4): p. 71:1–71:37.

Hu, C., W. Gao, C. Xu and K. Ben. 2020. Study on the Application of Digital Twin Technology in Complex Electronic Equipment. In *Proceedings of the Seventh Asia International Symposium on Mechatronics,* edited by B. Duan, K. Umeda and W. Hwang, p. 123–137. Lecture Notes in Electrical Engineering. Springer Singapore.

Hu, Y., C. Liu, M. Zhang, Y. Jia and Y. Xu. 2022. A Conceptual Framework of Cyber-Physical Remanufacturing System. In *nanoMan 2022 and AETS 2022 - 2022 8th International Conference on Nanomanufacturing and 4th AET Symposium on ACSM and Digital Manufacturing*. IEEE.

Janabi-Sharifi, F., L. Deng and W. J. Wilson. 2011. Comparison of Basic Visual Servoing Methods. *IEEE/ASME Transactions on Mechatronics* 16 (5): p. 967–983.

Jorg, S., J. Tully and A. Albu-Schaffer. 2014. The Hardware Abstraction Layer — Supporting control design by tackling the complexity of humanoid robot hardware. In *2014 IEEE International Conference on Robotics and Automation (ICRA),* p. 6427–6433. IEEE.

Kächele, F. 2023. Developments in Theory and Application of Copulas: Variance Estimation, Economic Modeling, Forecasting and Machine Learning. Karlsruhe, Germany: KIT Scientific Publishing.

Kadner, S., J. Kobus, E. G. Hansen, S. Akinci, P. Elsner, C. Hagelüken, M. Jaeger-Erben, M. Kick, A. Kwade, T. Müller-Kirschbaum, C. Kühl, D. Obeth, K. Schweitzer, M. Stuchtey, T. Vahle, T. Weber, P. Wiedemann, H. Wilts and R. Wittken. 2021. Circular Economy Roadmap for Germany. Technical report. acatech - Deutsche Akademie der Technikwissenschaften.

Kaiblinger, A. and M. Woschank. 2022. State of the Art and Future Directions of Digital Twins for Production Logistics: A Systematic Literature Review. *Applied Sciences* 12 (2): p. 669.

Kaiser, J.-P., S. Lang, M. Wurster and G. Lanza. 2022. A Concept for Autonomous Quality Control for Core Inspection in Remanufacturing. *Procedia CIRP* 105:p. 374–379.

Kalaitzakis, M., B. Cain, S. Carroll, A. Ambrosi, C. Whitehead and N. Vitzilaios. 2021. Fiducial Markers for Pose Estimation. *Journal of Intelligent & Robotic Systems* 101 (4): p. 71.

Karabulut, E., S. F. Pileggi, P. Groth and V. Degeler. 2024. Ontologies in digital twins: A systematic literature review. *Future Generation Computer Systems* 153:p. 442–456.

Ke, C., X. Pan, P. Wan, Z. Huang and Z. Jiang. 2023. An Intelligent Redesign Method for Used Products Based on Digital Twin. *Sustainability* 15 (12).

Kerin, M., N. Hartono and D. Pham. 2023. Optimising remanufacturing decision-making using the bees algorithm in product digital twins. *Scientific Reports* 13 (1).

Kerin, M., D. Pham, J. Huang and J. Hadall. 2023. A generic asset model for implementing product digital twins in smart remanufacturing. *International Journal of Advanced Manufacturing Technology* 124 (9): p. 3021–3038.

Kerin, M. and D. T. Pham. 2019. A review of emerging industry 4.0 technologies in remanufacturing. *Journal of Cleaner Production* 237.

Kritzinger, W., M. Karner, G. Traar, J. Henjes and W. Sihn. 2018. Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine* 51 (11): p. 1016–1022.

Kurilova-Palisaitiene, J., E. Sundin and B. Poksinska. 2018. Remanufacturing challenges and possible lean improvements. *Journal of Cleaner Production* 172:p. 3225–3236.

Lee, C.-M., W.-S. Woo and Y.-H. Roh. 2017. Remanufacturing: Trends and issues. *International Journal of Precision Engineering and Manufacturing-Green Technology* 4 (1): p. 113–125.

Leven, F. and M. Heizmann. 2022. Influence of undetected corneal glints on gaze estimation during manual work. *tm - Technisches Messen* 89 (1): p. 43–48.

Li, S., Y. You, P. Zheng, X. Wang and L. Wang. 2024. Mutual-cognition for proactive human–robot collaboration: A mixed reality-enabled visual reasoning-based method. *IISE Transactions* 56 (10): p. 1099–1111.

Mangold, S., E. Gerlitz, S. Zimprich, M. Friedmann, S. Matthiesen and J. Fleischer. 2022. Systematic quantitative investigation of the unscrewing process with regard to breakaway torque. *Journal of Remanufacturing* 13:p. 52–66.

Matsumoto, M., S. Yang, K. Martinsen and Y. Kainuma. 2016. Trends and research challenges in remanufacturing. *International Journal of Precision Engineering and Manufacturing-Green Technology* 3 (1): p. 129–142.

Mayer, S. and K. Furmans. 2010. Deadlock prevention in a completely decentralized controlled materials flow systems. *Logistics Research* 2 (3-4): p. 147–158.

Minerva, R., G. M. Lee and N. Crespi. 2020. Digital Twin in the IoT Context: A Survey on Technical Features, Scenarios, and Architectural Models. *Proceedings of the IEEE* 108 (10): p. 1785–1824.

Monostori, L., B. Kádár, T. Bauernhansl, S. Kondoh, S. Kumara, G. Reinhart, O. Sauer, G. Schuh, W. Sihn and K. Ueda. 2016. Cyber-physical systems in manufacturing. *CIRP Annals* 65 (2): p. 621–641.

Nasr, N. and M. Thurston. 2006. Remanufacturing: A key enabler to sustainable product systems. *Rochester Institute of Technology* 23.

Navarro, S. E., D. Weiss, D. Stogl, D. Milev and B. Hein. 2014. Tracking and Grasping of Known and Unknown Objects from a Conveyor Belt. In *ISR/Robotik 2014; 41st International Symposium on Robotics,* p. 1–8.

Newcombe, R. G. 1998. Interval estimation for the difference between independent proportions: comparison of eleven methods. *Statistics in Medicine* 17 (8): p. 873–890.

NVIDIA Cooperation. 2024. NVIDIA Isaac Sim. https://developer.nvidia.com/isaac-sim. [Accessed: 08-May-2024].

Orawo, L. A. 2021. Confidence Intervals for the Binomial Proportion: A Comparison of Four Methods. *Open Journal of Statistics* 11 (5): p. 806–816.

Ortenzi, V., A. Cosgun, T. Pardi, W. P. Chan, E. Croft and D. Kulić. 2021. Object Handovers: A Review for Robotics. *IEEE Transactions on Robotics* 37 (6): p. 1855–1873.

Parker, D., K. Riley, S. Robinson, H. Symington, J. Tewson, K. Jansson, S. Ramkumar and D. Peck. 2015. Remanufacturing market study. United Kingdom: European Remanufacturing Network (ERN).

Pati, M. and U. Majumdar. 2020. A Letter on Belt Conveyor System as a Mode of Transportation in Industry. *International Journal of Research in Engineering, Science and Management* 3 (12): p. 75–79.

Patki, N., R. Wedge and K. Veeramachaneni. 2016. The Synthetic Data Vault. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA),* p. 399–410.

Peng, C., F. Xia, M. Naseriparsa and F. Osborne. 2023. Knowledge Graphs: Opportunities and Challenges. *Artificial Intelligence Review* 56 (11): p. 13071–13102.

Plattform Industrie 4.0. 2022. Details Of the Administration Shell - Part 1: The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC02). Technical report. Federal Ministry for Economic Affairs and Energy.

Pont, F., S. Kolski and R. Siegwart. 2005. Applications of a real-time software framework for complex mechatronic systems. In *Advanced Intelligent Mechatronics. Proceedings, 2005 IEEE/ASME International Conference on,* p. 1324–1329. IEEE.

Qi, Q., F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang and A. Nee. 2021. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems* 58:p. 3–21.

Redelinghuys, A. J. H., A. H. Basson and K. Kruger. 2020. A six-layer architecture for the digital twin: a manufacturing case study implementation. *Journal of Intelligent Manufacturing* 31 (6): p. 1383–1402.

Rocca, R., P. Rosa, C. Sassanelli, L. Fumagalli and S. Terzi. 2020. Integrating virtual reality and digital twin in circular economy practices: A laboratory application case. *Sustainability* 12 (6).

Sebanz, N., H. Bekkering and G. Knoblich. 2006. Joint action: bodies and minds moving together. *Trends in Cognitive Sciences* 10 (2): p. 70–76.

Shafto, M., M. Conroy, R. Doyle, E. Glaessgen, C. Kemp, J. LeMoigne and L. Wang. 2012. Modeling, simulation, information technology & processing roadmap. *National Aeronautics and Space Administration* 32 (2012): p. 138.

Shrivastava, A., S. Mukherjee and S. Chakraborty. 2021. Addressing the challenges in remanufacturing by laser-based material deposition techniques. *Optics and Laser Technology* 144.

Souza, V., R. Cruz, W. Silva, S. Lins and V. Lucena. 2019. A Digital Twin Architecture Based on the Industrial Internet of Things Technologies. In *2019 IEEE International Conference on Consumer Electronics (ICCE),* p. 1–2. IEEE.

Stączek, P., J. Pizoń, W. Danilczuk and A. Gola. 2021. A Digital Twin Approach for the Improvement of an Autonomous Mobile Robots (AMR's) Operating Environment—A Case Study. *Sensors* 21 (23).

Steindl, G., M. Stagl, L. Kasper, W. Kastner and R. Hofmann. 2020. Generic Digital Twin Architecture for Industrial Energy Systems. *Applied Sciences* 10 (24): p. 8903.

Steinhilper, R. 1998. Remanufacturing: The ultimate form of recycling. Stuttgart: Fraunhofer-IRB-Verl.

Sun, X., H. Yu and W. Solvang. 2023. A Digital Reverse Logistics Twin for Improving Sustainability in Industry 5.0. Edited by A. E., R. A., S. J.O., von Cieminski G. and R. D. *IFIP Advances in Information and Communication Technology* 690 AICT.

Sundin, E. and H. M. Lee. 2012. In what way is remanufacturing good for the environment? In *Design for Innovative Value Towards a Sustainable Society,* edited by M. Matsumoto, Y. Umeda, K. Masui and S. Fukushige, p. 552–557. Dordrecht: Springer Netherlands.

Tao, F., M. Zhang, Y. Liu and A. Nee. 2018. Digital twin driven prognostics and health management for complex equipment. *CIRP Annals* 67 (1): p. 169–172.

Teixeira, E. L. S., B. Tjahjono, M. Beltran and J. Julião. 2022. Demystifying the digital transition of remanufacturing: A systematic review of literature. *Computers in Industry* 134:p. 103567.

Tiwari, D., J. Miscandlon, A. Tiwari and G. W. Jewell. 2021. A Review of Circular Economy Research for Electric Motors and the Role of Industry 4.0 Technologies. *Sustainability* 13 (17): p. 9668.

Tolio, T., A. Bernard, M. Colledani, S. Kara, G. Seliger, J. Duflou, O. Battaia and S. Takata. 2017. Design, management and control of demanufacturing and remanufacturing systems. *CIRP Annals* 66 (2): p. 585–609.

Tozanli, O., E. Kongar and S. Gupta. 2020. Evaluation of waste electronic product trade-in strategies in predictive twin disassembly systems in the era of blockchain. *Sustainability* 12 (13).

UNEP. 2020. Global Resources Outlook 2019: Natural Resources for the Future We Want. New York: United Nations.

Vachálek, J., D. Šišmišová, P. Vašek, I. Fiťka, J. Slovák and M. Šimovec. 2021. Design and Implementation of Universal Cyber-Physical Model for Testing Logistic Control Algorithms of Production Line's Digital Twin by Using Color Sensor. *Sensors* 21 (5): p. 1842.

VanDerHorn, E. and S. Mahadevan. 2021. Digital Twin: Generalization, characterization and implementation. *Decision Support Systems* 145.

Wang, X. and L. Wang. 2019. Digital twin-based WEEE recycling, recovery and remanufacturing in the background of Industry 4.0. *International Journal of Production Research* 57 (12): p. 3892–3902.

Wang, Y., S. Wang, B. Yang, L. Zhu and F. Liu. 2020. Big data driven Hierarchical Digital Twin Predictive Remanufacturing paradigm: Architecture, control mechanism, application scenario and benefits. *Journal of Cleaner Production* 248.

Wei, S., O. Tang and E. Sundin. 2015. Core (product) Acquisition Management for remanufacturing: a review. *Journal of Remanufacturing* 5 (1).

Widmer, T., B. Tjahjono and M. Bourlakis. 2018. Defining value creation in the context of circular PSS. *Procedia CIRP* 73:p. 142–147.

Wiesner, S., L. Egbert and A. Zitnikov. 2022. Using Operational Data to Represent Machine Components Health and Derive Data-Driven Services. Edited by R. D. Kim D.Y. von Cieminski G. *IFIP Advances in Information and Communication Technology* 664 IFIP.

Wollschlaeger, M., T. Sauter and J. Jasperneite. 2017. The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0. *IEEE Industrial Electronics Magazine* 11 (1): p. 17–27.

Wu, C., X. Bi, J. Pfrommer, A. Cebulla, S. Mangold and J. Beyerer. 2023. Sim2real Transfer Learning for Point Cloud Segmentation: An Industrial Application Case on Autonomous Disassembly. In *Proceedings, 2023 IEEE Winter Conference on Applications of Computer Vision: WACV,* p. 4520–4529. IEEE.

Wurster, M., B. Häfner, D. Gauder, N. Stricker and G. Lanza. 2021. Fluid Automation — A Definition and an Application in Remanufacturing Production Systems. *Procedia CIRP* 97:p. 508–513.

Yang, S., A. M. R., J. Kaminski and H. Pepin. 2018. Opportunities for Industry 4.0 to Support Remanufacturing. *Applied Sciences* 8 (7): p. 1177.

Yang, Y., G. Yuan, J. Cai and S. Wei. 2021. Forecasting of disassembly waste generation under uncertainties using digital twinning-based hidden markov model. *Sustainability* 13 (10).

Zacharaki, A., T. Vafeiadis, N. Kolokas, A. Vaxevani, Y. Xu, M. Peschl, D. Ioannidis and D. Tzovaras. 2021. RECLAIM: Toward a New Era of Refurbishment and Remanufacturing of Industrial Equipment. *Frontiers in Artificial Intelligence* 3.

Zheng, Y., S. Yang and H. Cheng. 2019. An application framework of digital twin and its case study. *Journal of Ambient Intelligence and Humanized Computing* 10 (3): p. 1141–1153.

Zhu, Y., J. Cheng, Z. Liu, Q. Cheng, X. Zou, H. Xu, Y. Wang and F. Tao. 2023. Production logistics digital twins: Research profiling, application, challenges and opportunities. *Robotics and Computer-Integrated Manufacturing* 84:p. 102592.