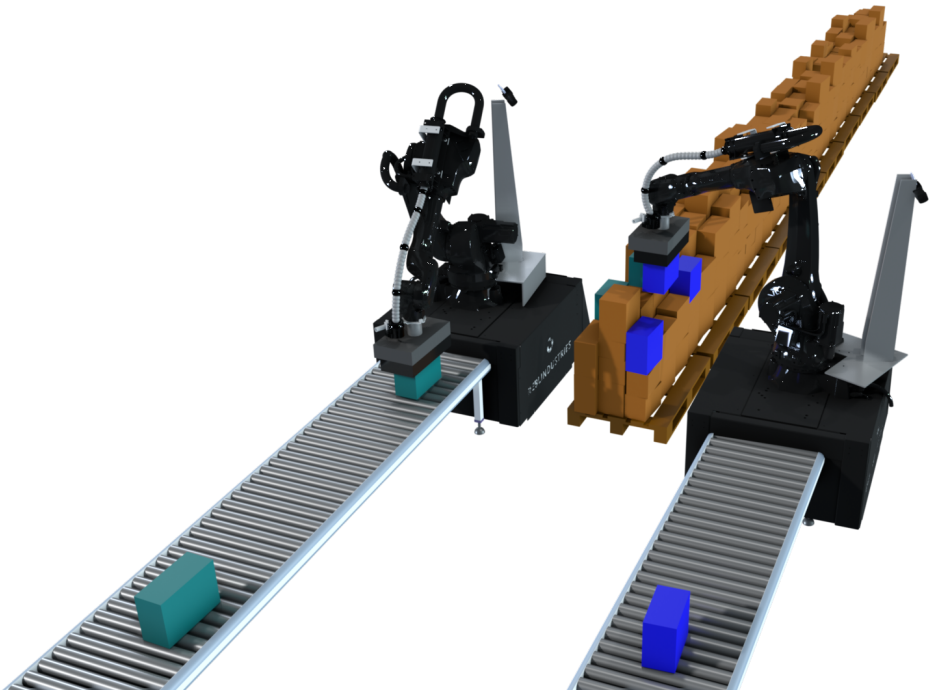
CHALMERS
UNIVERSITY OF TECHNOLOGY



Simulation-Based Optimization Study Evaluating Future Automation Potential in the Logistics Industry

Strategies for multiple mobile manipulators working in a shared space

Master's thesis in Production Engineering

KARL JANESTAD

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

Master's thesis 2024

Simulation-Based Optimization Study Evaluating Future Automation Potential in the Logistics Industry

Strategies for multiple mobile manipulators working in a shared space

KARL JANESTAD

Department of ELECTRICAL ENGINEERING
Division of System and control
Chalmers University of Technology
Gothenburg, Sweden 2024

Simulation-Based Optimization Study Evaluating Future Automation Potential in the Logistics Industry
Strategies for multiple mobile manipulators working in a shared space
KARL JANESTAD

© KARL JANESTAD, 2024.

Supervisor: Patrik Bergagård, Rebl Industries
Examiner: Martin Fabian, Department Electrical Engineering

Master's Thesis 2024
Department of Electrical Engineering
Division of System and Control
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A simulation model of two mobile manipulators working within a shared workspace.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Simulation-Based Optimization Study Evaluating Future Automation Potential in the Logistics Industry
Strategies for multiple mobile manipulators working in a shared space
KARL JANESTAD
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Simulation plays a crucial role in evaluating and analyzing the automation industry. With advancements in technology, particularly in AI and vision with object recognition, there is a growing interest in developing new solutions. This thesis investigates the current capabilities utilizing the simulation software KUKA.Sim. Additionally, it aims to explore the understanding of the potential applications and challenges of deploying multiple mobile manipulator robots (MMR) in a shared space. A simulation model with several new components was developed and tested using a simulation-based optimization methodology. To see the effects of different parameters in a dynamical chaining environment. A literature study and a current state analysis were made. One case study was identified and the data was used to compare the results from the iterative simulation process. The research findings show that it is possible to enhance the current robot simulation software KUKA.Sim and thereby close the gap between a simulation and a real-world environment. Furthermore, the test result of the iterative simulation processes revealed several important findings regarding how the camera placement affected the output and the importance of considering collision avoidance when using multiple MMRs in a shared space. Through the use of simulation, the logistics industry can effectively optimize new types of automation solutions, leading to an increased adoption of multiple MMRs.

Keywords: Simulation-based optimization, Task planning, Perception, Collision Avoidance

Acknowledgements

I would like to thank Martin Fabian for acting as both the supervisor and examiner for the weekly reconciliation of the project and the advice along the way.

Furthermore, a big thanks to the company Rebl Industry for their cooperation and help in making this research project possible. A special thanks to Patrik Bergagård for all the help in the technical aspect of the project and a big thanks to Johan Andersson for supporting with his knowledge and help in the state analysis of the industry.

Karl Janestad, Gothenburg, June 2024.

List of Acronyms

Below is a list of acronyms that have been used throughout this thesis listed in alphabetical order:

AGV	Automated Guided Vehicles
API	Application Programming Interface
ATLS	Automated Truck Loading Systems
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CPG	Consumer Packaged Goods
DES	Discrete Event Simulation
KPI	Key Performance Indicator
KRL	Kuka Robot Language
MMR	Mobile Manipulator Robot
NDA	Non-Disclosure Agreement
ROI	Return on Investment
UR	Universal Robots

Contents

List of Acronyms ix

List of Figures xv

List of Tables xvii

1 Introduction 1
1.1 Background 1
1.2 Stakeholders and Customers 2
1.3 Purpose and Outline of the Study 2
1.4 Research Questions 2
1.5 Delimitations 3

2 Theory 5
2.1 Current State in the Industry 5
2.1.1 Study Visit at DFDS 5
2.1.2 Interview with Rebl Industries Head of Concept Development 5
2.1.3 Industrial State-of-the-Art Practices 6
2.2 Simulation-Based Optimization 6
2.2.1 Di erent Types of Simulation Models 7
2.2.2 Robot Simulation Software 7
2.2.3 Digital Twin Within Robot Simulation 8
2.3 Perception in Simulation 8
2.3.1 Vision 9
2.3.2 Shared versus Separated Vision 9
2.4 Task Planning in Pick-and-Place Solutions 9
2.4.1 Collision Avoidance 11
2.4.2 Deadlocks 11

3 Methodology 13
3.1 Literature Study 13
3.2 Current State Analysis 14
3.3 Case Study Bring 14
3.4 Simulation Framework 15
3.5 Simulation Methodology 16
3.5.1 Data Collection with the Plugin Test-Leader 17
3.5.2 Statistical Analysis 17

3.6 Experimental Design 18
3.6.1 Strategies 19
3.6.2 Variants 20
3.6.3 Variations 21

4 Simulation Framework and Implementation of Simulation Model 25
4.1 Simulation Implementation 25
4.2 Developed Components 28
4.2.1 Component Ray-Cast 28
4.2.2 Component Signal-connector 29
4.2.3 Component Volume-Filler 29
4.2.4 Robot Code 30
4.2.5 Zone-Booking Implementation 30

5 Results 33
5.1 Test Results from the Iterative Process 33
5.1.1 Iteration of Parameters Using One MMR 33
5.1.2 Iteration of Parameters Using Two MMRs 37
5.2 Reference Data: Case Study Bring 39

6 Discussion 41
6.1 Simulation Framework 41
6.1.1 Development of the Simulation Components 41
6.1.2 Development of the Robot Code 42
6.1.3 Challenges and Problems with the Simulation Model 42
6.1.4 Simpli cations and Assumptions 43
6.2 Critical Analysis of Parameters in the Iterative Process 43
6.2.1 Disregarded Developed Implementations 43
6.2.2 Pick Strategy 44
6.2.3 Scan Strategy 44
6.2.4 Objects 44
6.2.5 Occurrences of Dropped Packages 45
6.2.6 Camera Placements 45
6.2.7 Focus Area 46
6.2.8 Arm Movement Strategy 46
6.2.9 Object Placement 46
6.2.10 Conveyor Placements 47
6.2.11 Constrained Space 47
6.3 Case Study Comparison 47
6.4 The Current State Analysis and Literature Study 48
6.5 Discrepancies: Simulation Versus Reality 48
6.6 Societal, Ecological, and Ethical Aspects 49
6.7 Recommendations and Further Research 49
6.8 Research Questions 51

7 Conclusion 53

Bibliography 55

A Appendix A I

B Appendix B III

C Appendix C V

List of Figures

2.1 The robot Stretch from Boston Dynamics [7]. 6
2.2 The NEURA Robotics solution for a mobile manipulator [8]. 6
2.3 Sequence of 30 picks from a heuristic approach [22]. 10
2.4 Sequence of 30 picks from a reinforcement learning algorithm [22]. . . 10

3.1 Picture of the simulation model representing the case study Bring. . . 15
3.2 The methodology used that was derived from the Banks model [12]. . 16
3.3 The three di erent sets of parameters used in the iterative process. . 18

4.1 Flowchart describing one cycle of movement. 25
4.2 Flowchart describing a complete simulation cycle. 26
4.3 Visualization of the simulation model where two MMRs pick-and-place packages onto their conveyors. 27
4.4 The Ray-cast component's parameterized values. 28
4.5 The developed Ray-cast component. 29
4.6 Package generated by the use of the Volume- ller component. 30
4.7 Visualization of the implemented zone-booking in the simulation model. 31

5.1 Visualization of how the camera placement was changed. 35
5.2 Example of the collected data from the case study Bring. 39

6.1 Visualization of packages excluded while picking. 46

List of Tables

3.1 The type of data collected during each simulation run. 18
3.2 The KPIs used in the research study. 18
3.3 The strategies, Variants, and Variations used in the iterative phase. . 19
3.4 The three phases used to test the di erent parameters. 23
3.5 Table illustrating how the parameters were set before a simulation run. 24
3.6 Illustration of extracted and saved data for each simulation run. . . . 24
3.7 An example of the calculated KPIs from the given data. 24

5.1 The saved and extracted data for the most interesting ndings in test phase 1, using one MMR. 34
5.2 The calculated KPIs for the most interesting ndings in test phase 1 using one MMR. 34
5.3 The saved and extracted data for the most interesting ndings in test phase 2, using one MMR. 36
5.4 The calculated KPIs for the most interesting ndings in test phase 2 using one MMR. 36
5.5 The saved and extracted data for the most interesting ndings in test phase 1 using two MMRs. 37
5.6 The calculated KPIs for the most interesting ndings in test phase 1 using two MMRs. 37
5.7 The saved and extracted data for the most interesting ndings in test phase 2 using two MMRs. 38
5.8 The calculated KPIs for the most interesting ndings in test phase 2 using two MMRs. 38

1

Introduction

As the automation industry expands into new areas, from more traditional use in the automotive sector, the degree of automation has risen in other sectors as well, including the eld of logistics. The demand has skyrocketed in the last couple of years in the logistics sector, where investments in service robots for goods and cargo grew by 44 % between 2021-2022 [1]. A marketing report by the company Style Intelligence, based on research and interviews with key players in the automation sector, revealed that the unloading and loading sector fundraising increased by thirtyfold between the years 2017-2023 [2]. There is a growing interest in the replacement of labor-intensive tasks performed by humans, given that these tasks are often perceived as monotonous, physically demanding, and time-consuming.

The increasing demand for the realization of these opportunities has led to the recognition of simulation-based optimization as an essential tool in the process of decision-making. Simulation-based optimization is a widely used term but refers in a more general case to the methodology to combine simulation techniques with optimization algorithms [3]. The use of simulation in production and manufacturing industries has been a well-established practice for several decades [4]. The adoption of simulation in the late 1990s marked a signi cant shift in the industry, with the technology becoming a more common tool in companies' implementation phase, especially in the automotive sector [4]. The use of simulation before implementation is nowadays common practice for veri cation, validation, and analysis to identify bottlenecks along with process failures and optimization of cycle times. However, the use of simulation before implementation becomes limited when more complex and dynamic environments are introduced. In response to these new challenges, there is a strong industrial interest to examine how to meet these demands. The objective of this thesis was to rst examine the current capabilities and limitations of simulation-based optimization in the robotic segment, with the use of multiple MMRs, and then based on the ndings recommendations were made.

1.1 Background

The research project was conducted in collaboration with the company Rebl Industries to develop a simulation model exploring potential applications in the logistics sector. Rebl Industries was established in 2018 and is a small start-up company

that focuses on automation as a service and specializes in robot applications with increased complexity [5]. The premise of the thesis was to use already established simulation methods to expand and investigate the use of MMRs, employing the created simulation model. The research used an experimental design, referred to as simulation-based optimization, to examine di erent parameters and strategies that come into play.

1.2 Stakeholders and Customers

The company Rebl Industries served as both the stakeholder and the customer, supervising the project and with the intent to use the solutions and recommendations derived from the research. The developed simulation model and ndings, along with the case study data, are the property of Rebl Industries, but have been approved for use in this paper. Individual-speci c data has not been included, as the assumptions used for comparisons with manual labour have been estimated based on other factors. Furthermore, the role of Chalmers University of Technology in the project was to act as a stakeholder, providing the framework and guidance for the academic work.

1.3 Purpose and Outline of the Study

The objective of the research was twofold. Firstly, the capabilities of simulation as a tool were explored. Secondly, the developed model investigated a speci c scenario in the logistics sector with di erent con gurations. The use of the simulated data and outcomes provided a comprehensive understanding of the potential applications and challenges of deploying multiple moving manipulators in a real-world environment. Through this exploration, key takeaways were found, and the ndings can serve as guidelines for further work.

1.4 Research Questions

Two research questions were de ned for the project. By utilizing simulation software in a simulation-based optimization methodology, along with a thorough current state analysis including literature studies, an interview, and a study visit, the following research questions were answered.

1. What strategies can be employed to address the utilization of multiple mobile manipulator robots, working collaboratively to achieve a common goal within the logistics industry?

2. To what extent and how can robot simulation, and in particular the software KUKA.Sim be utilized to examine research question 1?

1.5 Delimitations

The initial idea of the project revolved around a specific potential use for Rebl Industries. Some segments, where the findings can potentially be applied later were intentionally excluded from the thesis work due to a signed Non-disclosure agreement (NDA), between the company Rebl Industries and the author of this report, which limits some of the discoveries to be included in the report. The delimitations of the project were to use a broader and more general case in the exploitative phase and apply the findings from the presented case study.

The use of only the specific simulation platform Kuka.SIM [6], was set as a delimitation that limited certain aspects of the work. Furthermore, the time of the approximately 20 weeks of work was a limited factor. As the project was undertaken by one person and involved a broad scope of simulation development, 3 milestones were set in the original thesis proposal Appendix C. A risk analysis was made at the beginning of the project along with a Gantt chart to keep track of the progress.

Furthermore, a lot of data considered had to be excluded for security reasons, where the data of the case study limited some of the interesting research findings. There was no empirical study done on the case study because of security and time limitations.

2

Theory

This chapter presents a review of the current state-of-the-art in the industry along with an explanation of the research study's main areas.

2.1 Current State in the Industry

In order gain a deeper insight into the current state of automation in the logistics industry a study visit was conducted. Furthermore, an interview was held with the head of concept development at Rebl Industries.

2.1.1 Study Visit at DFDS

The study visit was arranged due to a potential automation solution that Rebl Industries would provide. In consideration of the con dential nature of the information, an NDA was signed, thereby limiting the information contained within this paper. However, the signi cant nding was that the logistics industry is behind in implementing automation solutions, and is mainly reliant on manual labor. Furthermore, the prevailing trend in automating warehouses is to automate selected parts of the process while other parts are overlooked. This results in bottlenecks, which reduces the overall e ect and can cause the larger investment to be redundant.

2.1.2 Interview with Rebl Industries Head of Concept Development

An interview was conducted with Johan Andersson, one of the co-founders of Rebl Industries, who currently holds the position of head of concept development. This interview was carried out to get a deeper understanding of the automation industry and the company's vision. The questions from the interview can be found in Appendix A. The interview revealed several important insights into the current state of the automation industry. It highlighted the potential of using new technologies, such as AI, to expand the possibilities of using robotics for the development of innovative solutions. For Rebl Industries this applies to the use of more advanced vision and object recognition capabilities.

2.1.3 Industrial State-of-the-Art Practices

One of the most used MMR solutions available today is the Boston Dynamics robot Stretch, Figure 2.1, released in 2023 [7]. This robot exempli es the capabilities that can be utilized in the logistics industry by having an MMR used for pick-and-place solutions. The manipulator is designed to be deployed without any training and can be integrated on-site for use in several types of work tasks. Another example is provided by the company NEURA Robotics [8], Figure 2.2. Their solution for a moving manipulator is a cognitive robot assistant with applications spanning from assembly to quality inspections. The NEURA manipulator is primarily designed for tasks requiring precision, with limited capability of lifting heavy objects. These two solutions are at the forefront of the pick-and-place segment, but in the interview with Johan Andersson he mentions that despite the high demand for these solutions, it has many cases not yet reached the desired maturity level.

The rapid evolution of technology and the diverse range of applications for mobile manipulators are driving an increasing number of companies to invest in the logistics sector. The logistics sector encompasses a wide range of activities related to the movement of goods. The term logistics can can encompass a range of meanings, in this context it refers speci cally to retail, e-commerce, and consumer packaged goods (CPG). The logistics industry encompasses a multitude of areas, it is often divided into four main categories: automated truck loading systems (ATLS), automated guided vehicles (AGV), yards, and the pick-and-place segment. The pick and place segment has attracted approximately 50% of the funding between 2015 and 2023 [9].

Figure 2.1: The robot Stretch from Boston Dynamics [7].

Figure 2.2: The NEURA Robotics solution for a mobile manipulator [8].

2.2 Simulation-Based Optimization

Simulation is used across various elds to support decision-making, enabling the prediction of product or process performance before implementation [4]. To utilize the

capabilities of simulation, a simulation-based optimization approach is commonly used [10]. This technique uses simulation experiments to evaluate the performance of a system in relation to the system variables. The use of simulation-based optimization allows for a systematic approach to optimization problems, enabling veri cation, validation, and analysis of results [10]. Using simulation and the methodology of simulation-based optimization compared to a traditional approach of more direct implementation can drastically reduce time and costs in the planning phase. This allows for experimentation with low cost and no risk in a virtual environment [4, 11].

2.2.1 Di erent Types of Simulation Models

Simulation softwares uses di erent types of simulation models. The two most used models in simulation are categorized as discrete event simulation (DES), and continuous simulation [12]. A model is de ned as a representation of a system used for studying that system [12]. The DES model uses discrete states, and the state variables change only at discrete time instances. DES is for example used in line optimization in a simulated production system, where each station has a de ned process time, and the output is presented in discrete steps. In a continuous model, the state variables change continuously over time, for example, if measuring the volume of a liquid coolant for a system, the liquid level continuously changes in response to the in ow and out ow of the coolant [12].

A simulation model for robot simulation can be viewed as both a DES and a continuous model. In the DES model context, it involves discrete steps triggering a new action after a certain amount of time. In contrast, the arm movement of the robot is calculated as an inverse kinematic movement and therefore is regarded as a continuous model. A simulation is therefore often not completely a discrete or continuous model, but since one type often dominates the model, it can be classi ed as one of them [12].

Simulation models are also further classi ed as being deterministic or stochastic, and static or dynamic. A simulation model that is only determined by its initial conditions and contains no random variables can be classi ed as deterministic. If the outcome of the model instead has some uncertainties, and the outcome can not be predicted, it is considered to be a stochastic model [12]. A static simulation is analyzed at a single point in time, for example, determining the stress distribution on a bridge. While a dynamic simulation represents a system as it changes over time [12].

2.2.2 Robot Simulation Software

There are di erent types of robot simulation software available, with the major robot companies often having their own software developed for their product catalog. For example, the company ABB has the robot simulation software Robot studio [13], and the company Kuka Group has the simulation software KUKA.Sim [6]. It is important to recognize that the Kuka Group already developed features for the use of MMRs and the simulation capabilities in parts such as navigation solutions.

There are several solutions on the market but often constrained to certain tasks and more focused on AGV solutions and bin picking [6].

The software employed by a company using robots is dependent on the hardware used, such as the physical robot itself. The reason for this is that using software speci c to the robot model allows the use of the robot-speci c programming language as well. As a consequence of this, it is common for companies to use only one software, given that it is not cost-e cient to have di erent robots with di erent programming languages [14].

Robot simulation software has improved signi cantly since the rst introduction in the 90s [4]. The simulation robot libraries are now often well-developed, and the libraries contain robot models representing a real physical robot in all aspects from kinematics to signal processing [11]. One of the research question focuses on the exploration of the software KUKA.Sim. This is a software used for the creation of 3D layouts with the KUKA.Sim libraries. The robot code can then be transferred to a real industrial robot system and tested under real-world conditions [11].

2.2.3 Digital Twin Within Robot Simulation

The term digital twin is often used in a broad spectrum of contexts, but can generally be refereed to as a virtual counterpart that utilize real-time data to mirror the behaviour of a physical counterpart [15]. Primarily, a digital twin is used in a simulation software to simulate the physical system under di erent conditions. This can be done by integrating real-time data and algorithms, as well as the geometric representation of the physical robot. In robot simulation, a digital twin represents the physical principle of the robot arm movement, which is used to predict and analyze simulated scenarios [15].

The geometric and physical representation of the hardware in a digital twin can be achieved with the use of computer-aided design (CAD), integrated into the simulation software. This allows for CAD les to be directly used in the software, for example, the robot itself or the camera used for the vision. This seamless transition from design to analysis enables the user to simulate and optimize models more precisely [15].

2.3 Perception in Simulation

From the literature study, and the interview with Johan Andersson (Section 2.1.2), the current state-of-the-art in robot simulation software's does not to any extent take into account the perception of objects, which in the context of simulation can be describe how something is understood and interpreted in a virtual environment. Therefore, a more general clari cation of perception and the use of vision is explained.

2.3.1 Vision

Object recognition is an important requirement in industrial robot applications using two-dimensional (2D) vision data for detecting three-dimensional (3D) objects [16]. The hardware consists of various types of cameras, with many di erent variants available on the market. One industry-leading camera that is used is the Photoneo motion camera [17]. It uses a convolutional neural network (CNN). The CNN is trained on data sets and can with a high accuracy detect objects and convert the data into a 2D grid, which is then translated into a coordinate system [17]. From the robot's frame of reference, its corresponding transformation matrix can be used with the given data to locate the position of the object. There are many di erent algorithms and methods available. One example used is the algorithm YOLO, which is a current state-of-the-art machine-learning algorithm [18], used for object recognition. With more advanced algorithms developed for vision capabilities in recent years, there are a lot of possibilities for new types of automation solutions.

2.3.2 Shared versus Separated Vision

The perception regarding the use of multiple MMRs in a shared space can be categorized into two distinct approaches, either sharing one vision system or using separate vision systems. In a separate vision system an individual camera is used for each MMR, like the Boston Dynamic robot Stretch [7], While the method of a shared vision system employs one camera for several MMRs. The use of a shared camera is often more common industry practice for stationary cells. A stationary cell in this context refers to a robot placed in a xed position that does not move within the cell where it is placed, unlike an MMR. This is usually a more easily solved problem, as a single source allows for simpler task planning [19]. The complexity in the case of several separate vision systems arises because the task planning will have to be carried out by distributed algorithms, instead of single-processor sequential algorithm [19].

2.4 Task Planning in Pick-and-Place Solutions

Task planning, in this study de ned as the scheduling of tasks in a certain order, has applications across various sectors in the automation industry. Commonly encountered challenges involve vehicle routing problems, and collaboration between multiple units. Task planning in robotics is used to guarantee optimal and collision-free paths [20]. One of the industry leaders in task planning is the Fraunhofer-Chalmers Research Centre for Industrial Mathematics, which utilizes state-of-the-art techniques for task planning with advanced mathematical models and algorithms, to compute optimal collision-free paths [20]. The methods are often used in production line balancing, where several robots have a cycled task, optimized to be done in the shortest time possible. A task and path planning solution by the Fraunhofer-Chalmers Research Centre is for example implemented at one of Volvo Cars AB manufacturing welding lines [20].

The task planning used in pick-and-place solutions di ers from the conventional stationary manufacturing processes. The stationary processes, such as the Volvo cars' welding line, use the same computed scheduling of tasks on the same type of car over and over again. In the interview with Johan Andersson 2.1.2, he mentions that the scheduling can become redundant in pick-and-place solutions in just a few seconds because of a dynamic and changing environment, which leads to errors and dropped objects. Therefore, the window of opportunity to use the computed schedule is small. More advanced algorithms also have longer computation time. The time frame for scheduling the tasks is less than a few seconds, which many of the more advanced scheduling algorithms surpass [21]. Furthermore, the ability to perceive all objects if they are obscured cannot be assumed, creating problems if the objects di er in size, since the pick position would only be possible to compute if the complete object is visible.

The research done of task planning within this paper of simulation-based optimization study evaluating future automation potential in the logistic industry, is limited to heuristic approaches, separated from the description of heuristic algorithms. Therefore, not all the available algorithms and methods will be explained. A heuristic approach, sometimes refereed to as classical task planning, uses practical, rule-based strategies to nd a solution [21]. This can for example be planning the sequence of tasks based on the shortest Euclidian distance to travel, or setting conditions such that the task planning sequence is determined by the greatest value in the de ned coordinate system. Figure 2.3, presents a common industry-used practice for pick-and-place task planning, by picking in a sequence in a top-down order. Figure 2.4 instead shows a optimized sequence as the total distance to travel is minimized, by a reinforcement learning algorithm [22].

Figure 2.3: Sequence of 30 picks from a heuristic approach [22].

Figure 2.4: Sequence of 30 picks from a reinforcement learning algorithm [22].

2.4.1 Collision Avoidance

Collision avoidance is closely related to task planning, depending on the strategy and methods used. When multiple MMRs are present in a shared space the potential for collisions among them must be considered [19]. This is either done by using a centralized unit to plan collision-free paths or by planning the paths of all the robots independently and then using a centralized unit to coordinate these paths. Centralized approaches are computationally demanding and rely on a global communication network [19].

Di erent methods exist to address collision avoidance, more traditional methods rely on zone-allocation to distribute the tasks accordingly. While more recent advancements use distributed algorithms by employing separate communication between pairs of robots [19]. The general idea of collision avoidance is some type of constraint given by either zones or distances. A more traditional method in collision avoidance is zone-booking, where zone-booking enables each robot arm movement to move only in a so-called booked zone [23]. Using a zone-booking approach, the zones are generated in a grid within the de ned pick area, with each zone marked as either booked or free. All the zones in the grid that are visited along the path to the pick position can then be found. When a pick sequence starts, a state with a calculated cost to reach each zone is computed, and depending on the cost, an action is given. The cost calculation can be made in various ways depending on the constraints that are set [23].

2.4.2 Deadlocks

With the use of collision avoidance, another problem occurs, namely deadlocks. A deadlock is a situation where two or more manipulators are unable to proceed because each manipulator is waiting for the other, resulting in a standstill [24]. This type of situation regarding several manipulators is commonly solved by interlock policies, where di erent methods and constraints are used to generate supervisors to ensure that the deadlock situations are avoided [25].

3

Methodology

This chapter introduces the case study and the methodology of the literature study, the simulation framework used, and the process of comparing various setups, and task and path-planning strategies.

3.1 Literature Study

At the start of the project, a thorough literature study was conducted to get an understanding of the subject. The main sources used were gathered from the research sites IEEE, ReearchGate, and Google Scholar. Because of the broad nature of the project's research questions, over 40 papers was examined.

The literature study was done by setting up four keywordssimulation-based optimization, task planning perception, and path planning. Each paper was then categorized into one of these four keywords. The path planning was later in the project renamed to collision avoidancefor the purpose of clarity. The defined keywords were then used as the search words on the research sites, and when an interesting and relevant paper was found, it was pasted into an Excel document under each corresponding keyword. During the literature study process, more search words were added as an understanding of the subject and different parts rose. The words added were mobile manipulator robot, vision, KUKA.Sim, heuristic task planning, object detection, deadlock, interlock, robot simulation, digital twin, and pick-and-place. The relevance of each paper was based on the criteria of what year it was published, how many citations the paper had, and how well it corresponded to the four categories defined. Technology advancement is rapid in all four categories, and therefore papers published in the last five years were mainly used, with some exceptions.

There were few papers in the literature study addressing the KUKA.Sim software or similar projects. Similar projects refer to exploring robot simulation in a simulation-based optimization approach. This corresponded to the Rebl Industries view of the exploratory nature of the project. To obtain sufficient knowledge and understanding of the state-of-the-art in the subject. The literature study was supplemented by the expertise and knowledge of Rebl Industries.

3.2 Current State Analysis

An interview was conducted with the head of concept development at Rebl Industries with the intent to create an understanding of the industry and to verify the claims that were found in the literature study. This was valuable as Rebl Industries is at the forefront of the pick-and-place segment in the logistics sector. Additionally, a study visit was conducted at the company DFDS, a shipping company that also provides services in the intralogistics sector [27]. The study visit further enhanced the understanding of the research study and the needs in the logistics sector.

3.3 Case Study Bring

The purpose of this thesis, conducted in collaboration with Rebl Industries, was to explore the future of automation in logistics with the use of MMRs. To achieve this, an already implemented project was identi ed, to have a current state-of-the-art overview of a functional solution, and to extract data for comparison. The project chosen for the case study involved a robot currently operating at a logistics distribution warehouse managed by the company Bring in the Jönköping area.

The project was chosen early on as a suitable case study project, due to the implemented cell has been running for several years and therefore a lot of data has been collected. The original implementation of the robot cell along with the working method of using simulation as validation before implementation, served as a good starting point to enhance and develop a simulation model for the research study, as the simulation basis and groundwork were already established. The pick-and-place cell solution located at Bring was developed by Rebl Industries to replace manual labor, transferring packages between a pallet and a conveyor. In the case study setup, the robot lifts the packages onto a conveyor, which transfers the packages to a main conveyor, to be dispatched to corresponding hubs for transportation to new destinations. The distribution warehouse functions as a central point for consolidating packages. When one pallet is empty, an operator replaces the pallet, and the cycle restarts. A visualization of the stationary cell located at Bring can be seen in Figure 3.1.

Figure 3.1: Picture of the simulation model representing the case study Bring.

3.4 Simulation Framework

The project evolved around setting up a simulation environment designed to explore new capabilities in the logistics industry. This was done by creating a base model that could be reused for future work by Rebl Industries. Rebl Industries use the KUKA.Sim simulation software [6]. This simulation software is a well-developed platform, where many of the necessary functions needed for the research study were already available. Even so, the research suggests the use of a vision system and the use of an iterative test methodology, both of which KUKA.Sim has limited capabilities.

The initial phase of the project along with the literature studies evolved around developing the base model, which could then be used for the iterative phase. To create a base model that represented the case study, several components had to be developed, and these were constructed using the built-in application programming interface (API), which in KUKA.Sim is based on the programming language Python. This was then used to build each speci c component that was needed for the base model. These components were constructed both for the speci c case of the study and for further use for Rebl Industries.

3.5 Simulation Methodology

To perform a thorough analysis using simulation-based optimization, the approach was to use the basis of the Banks model [12]. The Banks model is a methodology developed for use in DES simulation. The model consists of the stages problem formulation, project plan, conceptual model, and data collection. By this approach, it was possible to build a base model and from that verify both the model and the developed code. After the base model had been verified, the experimental design began. The loop in the experimental design, Figure 3.2, steps Experimental design and Production runs and analysis, consisted of the iterative process to change the defined parameters strategies, variants, and variations, described in Section 3.6.3.

Figure 3.2: The methodology used that was derived from the Banks model [12].

3.5.1 Data Collection with the Plugin Test-Leader

To analyze the output data that is given from the KUKA.Sim software Rebl Industries have developed a plug-in, named Test-Leader. This plug-in was used to collect the results of each simulation run during the iterative process. The data was then saved into an Excel le. The data that is referred to is the output log of each MMR. With the use of implemented counters and timers, it was possible to monitor and save the corresponding data. An example of the output log as the Excel spreadsheet can be found in Appendix B.

3.5.2 Statistical Analysis

The type of data that was collected from the Test-Leader during each iteration run is shown in Table 3.1. Each type of data in the table can be described as follows: The Number of packages picked was de ned as the total number of packages that were picked by each MMR during one simulation run. The total simulation time was the time from the start of the simulation until either the prede ned condition of the max number of packages picked was met, or no more packages were found in the scene. The Time for Manipulator-Motion was given as only the time the robot arm was moving, while the Time for Wait-Sec was the time given only when the robot code was executing a command. For example, in the Time for Wait-Sec, if the camera was triggered with a scan it took around 1000 ms in the simulation for the package list to be calculated and sent back before the MMR started to move. The Time for Wait-For was given as the time when conditional functions were used. For example, if a wait-for statement was used to wait for another MMR to move out of the booked zone. The number of scans represents the total number of scans performed on each MMR during one simulation run. A scan was de ned as one sequence where the camera component received a signal from the MMR to execute a scan, capture the packages within the view eld, generate a pick list, and send it to the MMR.

Only one value from the simulation data was tampered with. This was the Time for Wait-Sec time. This value was reduced in regards to how many scans were made. The reason for this was the simulation model's capabilities considering the signal processing time. With the implemented camera component, the simulation's Time for Wait-Sec time was much longer than the realistic processing time of a scanner. The total Wait-Sec time was therefore reduced by subtracting the number of scans multiplied by 3. The reason for these values was that the processing time of a scanner can be considered to take around 1000 ms, referenced from the interview, Appendix A, and the average processing time of each scan in the simulation took around 4000 ms.

Table 3.1: The type of data collected during each simulation run.

Data Collected
Number of packages picked
Total simulation time [s]
Time for Manipulator-Motion [s]
Time for Wait-Sec [s]
Time for Wait-For [s]
Number of scans

From the data collected during each iteration run, four KPIs were calculated, Table 3.2. The KPIs that were used in the simulation-based optimization study correspond to the KPIs from the case study Bring. However, one KPI from the case study, Pallet Duration, was excluded as the research study strictly focused on the pick-and-place output.

Table 3.2: The KPIs used in the research study.

KPIs
Average amount of packages picked per hour
Average picking time per package
Packages not placed
Percentage of failed picks

3.6 Experimental Design

After the base model was validated and veri ed, the experimental design began and an iterative process between three di erent sets of parameters was conducted. The sets of parameters were de ned as strategies, variants, and variations, Figure 3.3.

Figure 3.3: The three di erent sets of parameters used in the iterative process.

Each set of parameters is listed in Table 3.3. The milestones for testing these parameters were established at the beginning of the project and are available in Appendix C.

Table 3.3: The strategies, Variants, and Variations used in the iterative phase.

Strategies	Variants	Variations
Pick strategy	Objects	Object placement
Zone-booking strategy	Constrained space	Occurrences of dropped packages
Scan strategy		Occurrences of segmentation error
Arm movement strategy		Camera placement
		Focus area
		Conveyor placement

3.6.1 Strategies

The strategies were de ned as the parameters that were set before the simulation run and impacted the decision-making of the cell.

Pick strategy

From the theory review of task planning, section 2.4, ve di erent heuristic task planning methods were de ned. The coordinate system was de ned as a positive z axis in the direction up from the horizontal plane, while the positive x axis was directed away from the MMR. A positive z value indicated the highest point of the packages in the pick list, and a positive x value indicated the package furthest away from the MMR. The ve heuristic task planning methods describe the order the packages were picked. The pick list was sorted in two steps. For example, if the rst sort was made on a positive z value, the pick sequence started from the top of the highest-located package. A second sorting was then made if the package's z height was within a speci ed range of 10 mm. The reason for this approach was that if the packages on the pallet had approximately the same height, 10 mm corresponding to the margin of error, the e ect on the pick sequence in the 2-dimensional x and y space could also be analyzed, given that the packages had approximately the same height.

1. First sorted on positive z value, secondly sorted on positive x value.
2. First sorted on positive z value, secondly sorted on positive y value.
3. First sorted on positive z value, secondly sorted on negative x value.
4. First sorted on positive z value, secondly sorted on negative y value.
5. Sorted on positive z value.

Zone-booking strategy

With the use of the implemented zone-booking in the model, two strategies were de ned in how the MMR chose to proceed if a zone necessary for its next pick task was already booked by the other MMR. Only the zone of the designated pick position was regarded in the zone-booking strategy, excluding the zones within which the arm moved to reach the pick position. This was considered in the arm movement strategy instead.

1. If a pick position was booked the MMR waited until it was unbooked.
2. If a pick position was booked the MMR moved on to the next package in the list.

Scan strategy

The scan strategy was de ned as the number of picks an MMR should do between two scans. Di erent strategies resulted in di erent cycle times because each scan resulted in a signal process time.

1. A re-scan was made after the number of packages observed in the previous scan had been picked.
2. A re-scan was made after a prede ned number of picks.
3. A re-scan was made after each pick.

Arm movement strategy

The arm movement strategy was de ned as how the arm traveled from the drop-o position to the pick position, and back to the drop-o position again. This parameter was only used when the previously described zone-booking method, 2.4.1, was running on the MMRs. The use of multiple MMRs in uenced the time if the arm moved directly to the pick position by a point-to-point (PTP) movement, as it then booked and occupied more zones. Or if the arm movement rst had to travel to a point outside the pallet and the de ned zones. Therefore, the strategy a ected the time depending on whether the MMR had to wait for the other MMR. The zone-booking implemented is described more in section 4.2.5.

1. The arm moved the shortest possible path to the package.
2. The arm moved by a via point to the destination, the via point was located outside the pick area of the pallet.

3.6.2 Variants

The variants were de ned as the dimension of the objects, and the dimensions of the constrained space.

Objects

The objects refer to the packages that were generated at the beginning of the simulation run and then picked. These packages were generated by a Python script, described in Section 4.2.3, to be set to any dimension in length, width, height, and rotation around the de ned z-axis. The reference values that were used were found in the case study Bring. The values were then randomized within the prede ned range. For example, if the height value range was set to 100, 200, or 300 mm. One of these three values could set the height of the package.

1. The dimensions of the generated packages were set to the length: 300, 350, or 400 mm, the width 185 mm, and the height 150, 200, 250, or 300 mm. The range in the rotation was set between 0 6.

2. The dimensions of the packages were set within the range of the length: 200-800 mm, the width 150-400 mm, and the height 100-500 mm.

Constrained space

The constrained space was de ned by a volume intervening with the reach of the manipulator. The volume represented some type of physical constraint, for example, a wall next to the pallet.

1. The MMR had no constraints in space except for the pallets with packages and the other MMR.

2. The space available for the MMR was constrained by a volume representing a physical object.

3.6.3 Variations

These parameters were set to in uence the simulation run by variations.

Object placement

The object placement parameter constrained the size of the generated volume. It de ned the length, width, and height of where the objects were placed. Given the case study, the length and width of a pallet were constrained to an EU-pallet size, 1200 800mm^2. The height was set with a range between 1000 1600 mm.

1. The default value of the object placement was set to a volume of 4000 800 1100mm^3.

2. The object placement volume was set within the range of the length: 2000-12000 mm, the width 400-800 mm, and the height 1000-2000 mm.

Occurrences of dropped packages

A function was implemented such that the simulation had a threshold value set before each simulation run to simulate a real-case scenario of occurrences of dropped packages. The threshold value ranged between 0 100 %. Where 0 % meant the function was not used and the MMR did not drop any packages. While 100 % meant that every package that was picked was also dropped. The threshold value was kept between 0 10 %, this reference data was given from the interview, Appendix A.

1. The probability of dropping a package was set to 0 %.
2. The probability of dropping a package was set in the range 0 10 %.

Occurrences of segmentation error

The segmentation error was de ned as the parameter that took into account the probability of a generated pick position not corresponding to the packages actual position. This parameter was used as it can occur in real-case scenarios that the image annotation gets it wrong in where the packages are placed. This function was implemented with the same method as occurrences of dropped packages, a threshold value was set before the start of each simulation. The threshold value was set as the percentage of the likelihood of segmentation error between 0 100 %. If a segmentation error occurred, the pick position was moved by a random value between 100 500 mm. These values were chosen as a simpli cation of the representation of generating the wrong pick position from a scan.

1. The probability of a segmentation error was set to 0 %.
2. The probability of a segmentation error was set in the range 0 10 %.

Camera placements

The camera placement was adjusted in terms of height and rotation around x, y, z.

1. The camera was placed at the height 2200 mm and the angles -57, 0, 37°.
2. The camera was placed at a height between 1500-2500 mm, and the angles ranged between 0-360°.

Focus Area

The focus area of the camera was de ned as the depth, width, and height of the camera's view eld. The recommended dimensions of the focus area was given from the Photoneo camera model MotionCam-3D Color L+ [28], as a depth of 2470 mm, a width of 3246 mm, and a height of 2328 mm. A step size was de ned regarding how many beams were sent out in the width and height dimension, elaborated in Section 4.2.1.

1. The focus area was set to $2470 \quad 3246 \quad 2328\,\text{mm}^3$, with a step size of 80.

2. The focus area was set with the range of the depth: 1500-3000 mm, the width 1500-4000 mm, and the height 1500-4000 mm.

Conveyor placement

The conveyor placement of each MMR was de ned as how the conveyor was positioned.

1. The conveyor was placed behind the MMR.

2. The conveyor was placed to the side of the MMR facing the pallet.

Iterative process

As there were many di erent parameters de ned that could be tested, it was important to test the parameters systematically, both individually and in combination with each other, to see how the output and KPIs changed.

To enable this, three iterative phases were de ned, shown in Table 3.4. The initial test phase was done to review if the parameters worked as intended. As described in Section 3.5, the experimental design in the iterative process was looped back in the process to evaluate and consider the simulation capabilities and the parameter in uences. Test phase 1 was done to review each parameter by itself to draw some rst conclusion of the e ect each parameter gave. In test phase 2 the parameters were tested in combination with each other.

Table 3.4: The three phases used to test the di erent parameters.

Iterative phase	Implementation
Initial test	Reviewed the model to see if the parameters worked as intended.
Phase 1	Each parameter was tested separately.
Phase 2	Parameters were tested combined with each other.

The order of the testing of the parameters did not a ect the result but gave a guideline for the most interesting ndings early on. It was important to reduce the number of iterations to the extent it was possible and by that save time. Each iteration was analyzed and the next parameter change was considered regarding the previous ndings. Before the start of each simulation run, the parameters were set, an example is shown in Table 3.5.

From each iteration, the output log from the plug-in Test-leader saved the data to an Excel le when the simulation run was done. An example of the data that was extracted from the Excel le is shown in Table 3.6.

Table 3.5: Table illustrating how the parameters were set before a simulation run.

Iteration nr:		
Parameter	Value of parameter	Details
Pick strategy	1	
Zone-booking strategy	1	
Scan strategy	1	
Arm movement strategy	1	
Objects	300 400, 185, 150 320 mm	L W H
Constrained spaces	1	
Object placement	4000 800 1100 mm^3	L W H
Occurrences of dropped packages	0 %	
Occurrences of segmentation error	0 %	
Camera placements	-56.8, 0, 36.4°	De ned angles A; B; C
Focus area	2470 3246 2324 mm^3	D W H
Conveyor placements	1	

Table 3.6: Illustration of extracted and saved data for each simulation run.

Iteration nr	Packages	Total Time [s]	Robot-Motion [s]	Wait-Sec [s]	Wait-For [s]	Num. of Scans	None Pick Pkg
1	50	415.696	267.540	142.300	5.856	8	0

Given the result from the plug-in Test-leader the KPIs were then calculated. An example of the calculated KPIs is given in Table 3.7.

Table 3.7: An example of the calculated KPIs from the given data.

Iter.	Test-leader	Average picking time per pkg	Pkg not placed	Percentage of failed picks
1	433	8.3	0	0

Each iteration was tested with 50 packages picked. This quantity was considered su cient to obtain an accurate portrayal of the results, yet not larger than needed to avoid unnecessary simulation time. A higher number of packages picked was also tested and the di erence in the KPI average picking time per package was calculated, compared to 50 packages with the exact same parameters. The di erence was seen to be around 0.6 1.1 %. For example, when picking 50 packages compared to 100 packages with the same parameter settings and model, the KPI of the average amount of packages picked per hour was 541.7 compared to 537.1. This was considered in regards to using the randomized value introduced in some parameters, as dropped packages and segmentation error. Even so, the reasoning and with over 40 iterations it was possible to still have con dence in the result. From the result obtained in all the iterations, the data collected was calculated into the de ned KPIs. The KPIs were analyzed and the most interesting ndings are presented in chapter 5.

4

Simulation Framework and Implementation of Simulation Model

This chapter describes how the simulation framework was considered with the developed and implemented base model. This chapter is included as an additional part of the result to comprehensively show the developed simulation model.

4.1 Simulation Implementation

The objective of the base model built in regards to the case study Bring was to move a specified number of packages placed on pallets to a conveyor belt, in the fastest time possible, with the use of MMRs. Each pick-and-place cycle consisted of only three states, where the start and end position in the first and last pick of the simulation, was defined as the MMRs home position. During all other picks except the first and last pick, the start and end positions corresponded to the conveyor drop-off position. This movement was cycled during each simulation run to pick all the assigned packages, see Figure 4.1.

Figure 4.1: Flowchart describing one cycle of movement.

The complete simulation cycle is shown in Figure 4.2. The data was saved into the Excel le with the plugin Test-leader. The Test-leader could then rerun the simulation, with changes made to the prede ned parameter within their respective range.

Figure 4.2: Flowchart describing a complete simulation cycle.

The movement of each MMR was developed as follows. The camera component snaps a photo as a scan, and the MMR gets the corresponding pick position from the pick list. The MMR moves in discrete steps back and forth and picks the assigned packages. Based on the scan strategy, the MMR moves back to its de ned base position when a new scan should be taken. If a scan is taken and no packages are observed the MMR moves forward and updates its base position. With this approach, the MMR can move forward until the last pallet. The movement forward was set in discrete steps and was de ned as the value of 1000 mm. The value of the discrete distance was at the beginning of the iterative process a parameter itself. However, as it had a small e ect as the MMR only updated its base position a few times each simulation run, it was excluded as a parameter. A momentary picture of the cell can be seen in Figure 4.3. The colors of the packages indicate each manipulator's assigned pick, where the manipulators, depending on the strategy moves to the packages in a certain de ned order given by the set parameters.

In Figure 4.3, it is possible to see the placement of the cameras. As the right MMR places a package, its camera faces the pallet with the packages. In the implemented code the camera is triggered after the packages are placed on the conveyor, depending on whether a new scan is initiated or not. The triggering of the camera was only made during the drop-o of the package onto the conveyor.

Figure 4.3: Visualization of the simulation model where two MMRs pick-and-place packages onto their conveyors.

4.2 Developed Components

The project relied on developing a more advanced perception integrated into the simulation environment in KUKA.Sim, to be able to locate objects in regards to the obscurity of the vision. As described in Section 2.3.1, the development of these components was necessary to close the gap between a simulated environment and the physical environment, to represent the dynamic behavior of a cell. The component developed for the camera was named Ray-cast. In order to achieve the experimental design with the iterations of the de ned parameters, two other components had to be developed. These components were named Volume- ller and Signal-connector. The Volume- ller component was developed for the use of generating packages, while the Signal-connector connects the signals between the Ray-cast camera and the MMRs.

All components have been developed using Python integrated into KUKA.Sim, as a built-in API. The code is not described further, since the developed code belongs to Rebl Industries. The components are instead brie y described as to how they were developed.

4.2.1 Component Ray-Cast

The Ray-cast component was built to replicate the Photoneo camera model MotionCam-3D Color L+ [17], introduced in Section 2.3.1. This speci c camera was chosen because it represents state-of-the-art technology for object recognition and is used in industry, Appendix A. The Ray-cast component was built with a parameterized focus eld, which can be set to the desired values of depth view, width of depth view, height of depth view, and number of steps, shown in Figure 4.4.

To represent the behavior of the physical camera the Ray-cast component uses a built-in function from the KUKA.Sim API, this function uses ray beams. In this context, a ray beam refers to the path of a line that passes from the camera lens to the designated position. The number of steps set in the parameterized component, shown in Figure 4.4, decides the amount of ray beams that are sent out from the camera. For each ray beam that hits the package, the total area of the package is calculated along with which side that is hit. A threshold variable of the area covered per side can then be set. Based on the de ned threshold value, it is possible to determine the coverage of a side to be considered pickable.

Figure 4.4: The Ray-cast component's parameterized values.