MASTER THESIS

Machine Learning Opportunities in a High-Mix and Low-Volume Bin-Picking Assembly Station

Learning by Watching an Operator

G.J.B. Bauwens
DC 2021.090
October 2021

DEPARTMENT OF MECHANICAL ENGINEERING

EINDHOVEN UNIVERSITY OF TECHNOLOGY

MASTER THESIS

Machine Learning Opportunities in a High-Mix and Low-Volume Bin-Picking Assembly Station

Learning by Watching an Operator

DC 2021.090

Author:	G.J.B. (Guus) Bauwens	
Supervisors:	Prof.dr.ir. I.J.B.F. (Ivo) Adan	(TU/e)
	Dr.ir. A.A.J. (Erjen) Lefeber	(TU/e)
	Dr. T. (Tim) Foreman	(Omron)
	MSc. M. (Mariya) Yurchenko	(Omron)
Committee:	Prof.dr.ir. I.J.B.F. (Ivo) Adan	(TU/e)
	Dr.ir. A.A.J. (Erjen) Lefeber	(TU/e)
	Prof.dr.ir. R.M. (Remco) Dijkman	(TU/e)
	MSc. M. (Mariya) Yurchenko	(Omron)

Eindhoven University of Technology
Mastertrack Manufacturing Systems Engineering
Department of Mechanical Engineering
Dynamics and Control
Eindhoven

October 18, 2021

Abstract

This thesis presents an overview of the potential of using machine learning on assembly instruction algorithms meant for human-operated assembly stations, in a high-mix and low-volume production environment. As such learning algorithms require sample occurrences of the process of interest, this production environment with limited sample occurrences and many new products to be assembled, introduces challenges. For research purposes, Omron developed a demonstration setup that allows for the assembly of simple products consisting out of coloured bolts that can be placed in a flat grid. Using a similarity factor in the products, based on common building blocks the bolts make, allows introducing repetition in products while still allowing much variability, allowing to statistically compare the assemblies based on these blocks' features like position and shape. In addition, a generalised assembly strategy analysis is proposed and researched which describes the assembly sequences of products on a global level. It exists out of a concise set of strategies operators can use to assemble a product, and can be applied to all product variants. Locally, a similar but stricter analysis is performed concerning the assembly order of the building blocks. Twelve operators have assembled products and a performance and strategy analysis has been performed over their bolt placement data. Statistical proof if using a certain global strategy correlates to assembly performance is lacking, but differences in operator assembly preferences are found. Local strategy preferences do vary between users, with observed differences in assembly time within operator data. This shows the potential of indicating to operators which assembly order to use, based on self-learning of an algorithm that uses operators' own and others' assembly data.

Keywords: high-mix, low-volume, pick-to-light, product assembly, assembly preference, assembly strategy, self-learning, correlation, operator, assembly performance

Summary

To collaborate into research concerning the optimisation of operator performance in product assembly on human-operated assembly stations for highly variable production, Eindhoven University of Technology and Omron have come together in the Fieldlab Flexible Manufacturing. These assembly stations have seen improvements over the years to improve operator working conditions, ease of use and, more recently, instruction optimisation. These instructions are, however, created manually. Ideally, the setup can learn to adapt instructions automatically to the operator skill level and product variations to optimise assembly speed and assembly correctness.

Omron has developed an assembly station meant for the production of simplified products for demonstration purposes. Assembly instructions are provided using a pick-to-light system. Products are of the type high-mix, low-volume, meaning repetition of the exact same product is rare. As the foundation of a learning model is data, which is lacking in a high-mix and low-volume environment for specific products, a similarity measure based on repeating shapes of components within the product is defined in this thesis. By grouping assembly data based on these components' features, sample sizes of identical repetitions within products increase, allowing statistical analyses to be performed.

By tracking placement times of the parts of the product, a clear overview of operator performance is known, both on a global product level and on component level. The aim of this research is to improve performance, which is defined as increasing operator assembly speed and decreasing assembly errors. The setup should be able to adapt the instructions given to operators by use of self-learning based on gathered data. If operators statistically differ in their performance of the components or products, the potential of these learning algorithms can be shown by correlating the performance to product features based on the layout, position, and type of components the product is made of. However, differences in the feature values do not seem to impact the assembly performance much based on data gathered.

Linking the assembly order of the product parts and components assembled by the operator to the product and component features, allows to define global operator assembly strategies. These can describe the operators' assembly preferences. These preferences are directional based, colour based and component based. While statistical proof of using a certain strategy correlates to assembly performance is lacking, differences in operator assembly preferences are found.

The components itself require assembly steps too, which are described by local strategies. A comparison between the resulting occurrences show differences between operators. By linking these strategies to performance, correlations have been found for some operators. As local strategies do vary between users, there is potential for operator recognition, which is desired by companies to link performance anonymously to the current working operator.

It can be concluded that, while differences are found between operators, many aspects of the assembly do not show statistical differences or correlations to performance. Yet, as the placement times of bolts lack many operations by the operator, possible extensions of the research with potential of extracting more information from operators that describe and influence assembly performance are presented. These extensions are based on extracting motion sequences from operators and more extensive similarity measures of products and assemblies based on graphs. A graph representation of the products would allow to find matches in (parts of the) products, which is a more universally adaptable approach than the currently used approach which requires pre-defined components.

Preface

This thesis presents my biggest work as a student, carried out in circumstances I never imagined. My initial plan was to go abroad and explore life and studying in a new environment, but the Covid-19 pandemic disrupted those plans completely. Instead of studying abroad, I started my graduation project from my own room in Eindhoven. Working on this thesis during the pandemic certainly came with some challenges, with my biggest worry being data, but luckily I was able to perform the necessary experiments at the Brainport Industries Campus. I learned many valuable lessons during the various phases of this project, from defining the research direction to writing the final report. Looking back, I am happy to present this thesis, of which I am proud.

This thesis would not be here in its final form without the assistance and help of many people. A big thank you goes out to Erjen Lefeber and Mariya Yurchenko, my direct supervisors of respectively Eindhoven University of Technology and Omron, as well as my main supervisor Ivo Adan and my external committee member Remco Dijkman. Erjen, thank you for our weekly meetings where I could discuss my problems, show a cool new graph, and got valuable feedback. Your structured approach helped me being structured in keeping a weekly log of my progress, which proved helpful in making arrangements and back tracking previous results or decisions. Mariya, thank you for helping me with technical details concerning the setup, making time to discuss results with me and giving me feedback from your perspective. It helped me to also focus my results on their business value, and not to solely show results because they are results. Thank you Ivo for the overall supervision of my graduation project and our update meetings, which helped to get all the results together and present them in a concise manner. This also taught me to present information with as goal to receive feedback instead of solely presenting information. Also a thank you to Tim Foreman, European R&D manager of Omron, for joining these meetings and giving your insights. Besides Tim and Mariya, also a thank you to Duc from Omron for helping with technical questions related to the Omron setup. Finally, again thank you Remco for joining my graduation committee as external member.

Due to the Covid-19 pandemic, the number of people at the Fieldlab Flexible Manufacturing were limited. Therefore, I am very grateful that so many of my friends and family were happy to come to the Brainport Industries Campus to assemble products for me, to allow me to gather data. In alphabetical order, thank you Anne, Bavo, Erjen, Ineke, Joris, Lotte, Luc, Niels, Nienke, Nikolaj, Peggy, Stan and Thomas for helping me out. Without you, I would not have had results to present. Concerning the Fieldlab Flexible Manufacturing, also thank you to Edwin for our talks and to Marijke concerning our discussion of interactions between humans and machines.

I would like to give a special thank you to Ineke, Lotte and Luc for our physical lunch breaks and lunch walks at university whenever possible. A special thanks to Ineke for being someone to talk to, discuss and listen to during our workdays in Gemini. Finally a thank you to Joris, Niels and Stan for our weekly digital coffee breaks that allowed us to still being able to talk, update each other and give each other feedback, even in these special times.

Guus Bauwens
Eindhoven, October 2021

Contents

Chapter 1

Introduction

When Ford revolutionised the assembly of mass-produced automobiles in the early 20^{th} century, by adapting assembly lines and optimising every single step in the assembly process, product assembly would never be the same again [1]. Nowadays, instead of a single model with a single motor type and at most some livery colour options, cars come in all sorts models, with various motor choices, interior options and various other customisation options. With such variability, the exact same model is not produced very often anymore in comparison to mass production. While a different colour will probably not influence the way the car should be assembled, this principle of high variability could in other products introduce significant changes in assembly which cannot easily be optimised anymore. Opportunities for the optimisation of assembly in this kind of production environment are the main topics explored in this research.

Eindhoven University of Technology, Fontys University of Applied Sciences, Bosch Rexroth, Omron Industrial Automation and others have collaborated in the Fieldlab Flexible Manufacturing located on the Brainport Industries Campus (BIC) to do research in, e.g., the mentioned highly variable production [2]. Omron Industrial Automation has developed a bin-picking assembly line demonstration setup with the goal of showing the potential and methods of how machines can learn from themselves and humans. The objective is to trigger engineers to think and imagine how implementations developed on the demonstration setup could be implemented in actual manufacturing lines. As part of the self-learning assembly lines demonstration project within the Fieldlab Flexible Manufacturing, Eindhoven University of Technology and Omron Industrial Automation started a collaboration project to further develop and do research related to the Omron demo setup [3].

Full automation to replace human labour has been suggested as a potential solution to high labour costs. However, complicated tasks like low-volume, high-variability production require advanced technology, while humans perform well in such high complexity tasks [4]. Therefore, Van Rhijn et. al. (2017) [4] present support and assistance methodologies for operators upon which this research aims to extend by exploring possibilities to improve human efficiency using self-learning. The remainder of this chapter is structured as follows. Section 1.1 introduces the Fieldlab Flexible Manufacturing setting in which this research is performed. Section 1.2 formulates the problem and section 1.3 introduces the state of practice of this field of study. Section 1.4 introduces the research objective and finally the report outline is presented in section 1.5.

1.1 Fieldlab Flexible Manufacturing

The Fieldlab Flexible Manufacturing is a collaboration project between companies and knowledge institutes and is part of the innovation program *Factory of the Future*. Within the Fieldlab Flexible Manufacturing, all parties work on assembly automation of highly customisable and low volume products [2]. Manufacturing systems are called flexible in case they can quickly adapt to expected and unexpected changes in the type and quantity of the product being manufactured [5].

Within the Fieldlab, Omron Industrial Automation, known for product and appliance expertise of technological advanced industrial automation appliances [6], is the leader of the work package *self-learning assembly lines*. This work package does research with the aim of realising a production system that can facilitate error-free assembly of highly variable products with a high complexity level.

To physically facilitate research, Omron Industrial Automation, from now on called Omron, developed the demonstration setup to emulate a manually operated assembly station. The setup is focused on human-centred assembly of highly variable and complex products. The collaborative robots that are present on the setup therefore only assist the operator and do not replace the operator.

1.2 Problem Formulation

Since the time of having instructions on paper, with no adaptions possible except changing the full instructions using operator feedback, much has changed. Developments in the digital revolution modernised assembly stations by introducing digital instructions. Currently, state-of-the-art assembly stations use projected instructions on the assembly platform for direct instructions on where to place components. Which parts and how many parts to pick are indicated too on the part storage bins itself. This kind of assembly is called pick-to-light assembly. The Omron setup uses a beamer to depict the instructions, consequently this variant of pick-to-light assembly is called pick-to-beamer.

Customisation and assembly-to-order are trends in manufacturing. By the use of automation, assembly stations can be adapted to allow for custom products. This production environment is called high-mix, low-volume. Omron developed the demonstration setup to allow producing products in this production setting. In a low-mix, high-volume production environment, data of a specific product can be gathered to improve the assembly in detail. This data is however not available in a high-mix, low-volume environment as the variability of products is high, and the occurrences of unique products low. The challenge faced is to develop an algorithm that can analyse the production data of different operators and products, and adapt assembly instructions to improve this performance, without the guarantee of having access to data of the exact product to be produced. Just using optimal assembly patterns of previous products is therefore not possible. These optimal assembly patterns are firstly not known, again due to the limited data, and secondly, it is not known in what circumstances and how an optimal assembly pattern of one product can be transformed to another non-equal product.

Therefore, this research aims to show whether assembly data of products in this high-mix, low-volume environment can be used to implement self-learning on the provided assembly station setup. The setup tracks operator actions using a camera that logs the placement of parts of the products. This allows showing the potential of using this data and the other data available about the product to create an operator profile upon which custom assembly instructions are given to the operator automatically. These assembly instructions could eventually be in verbal form, in written text form, or visualised using the pick-to-light system installed on the setup.

1.3 Literature Study

The literature study can be divided into the state-of-practise and state-of-the-art. The state-of-practise covers previous work in the the field of bin-picking assembly and pick-to-light assembly, and state-of-the-art covers related research into assembly optimisation.

1.3.1 State-of-Practise

Previous research within the Fieldlab Flexible Manufacturing, worked towards optimisation of the task allocations between a human and robot agent in a flexible manufacturing assembly setup [7]. It focused on predicting the expected processing times of each assembly step and on taking into account sequential operation conditions. This approach made it possible to optimise the order of assembly and optimise the division of tasks between a robot and a human. Also, by comparing predicted effective process times per assembly step with data of specific operators, operators could be recognised, or a new operator could be automatically added to the system. The use-case explored in this research, however, focuses on unique products with unknown assembly times. If it can be shown that the assembly times of unseen products can be predicted based on previous data of other, different type of, products, a similar optimisation could be performed. However, it should be noted that the Omron setup and the product type are currently not suited for a similar kind of parallel assembly by an operator and a robot where they both work on the same product at the same time.

Previous research into the Omron setup resulted into a learning model that predicts optimal assembly recipes using decision trees [8]. Based on certain features of assembly instruction steps, the learning model can decide which set of assembly steps is optimal for a specific operator and, therefore, potentially gives different operators different assembly instructions. The predictors are:

- the number of steps of the assembly instruction;
- the average number of screws to be placed in an assembly step;
- the standard deviation of the number of screws per assembly step;
- the average number of different colours to be placed per assembly step;
- the number of times an operator has assembled the product type on the same day.

Every assembly step instructs to place a certain number of product components on the correct place. The number of product components per assembly step may be varied. The predictors represent a summary of the layout of every assembly step, as they are either a number, a mean value or a standard deviation. All predictors are computed using the product layout and assembly steps, statistical operations based on the layout and assembly steps, and logged operator operations. The response is assembly time. Prediction of assembly errors as a response of the learning model are not mentioned, but due to the nature of the task, assembly using a high assistance level based on pick-to-light, and based on experience with the setup, it can be assumed that assembly errors were uncommon.

The most important predictor turned out to be the number of times an operator had assembled the product. This does not give information about the way instructions should be presented to operators, as it originates from experience. The number of assembly steps in the assembly instructions was also correlated to the performance. The amount of data required for analysis is also considered by Stellas [8]. When multiple assembly instruction sets are used for a set of to be assembled products, data can get too scarce to make valid conclusions for all combinations of variables in the research. This is caused by these different assembly instructions having different values for the predictors, which is referred to as the *curse of dimensionality*, since the predictor dimension gets too large compared to the number of samples available [9]. In short, it is desired to limit the variability for testing purposes and to validate the performance over time within the high-mix, low-volume use case. Therefore, the research presented in this thesis deals with the trade-off to consider high-mix products, while limiting this variability too such that conclusions can be drawn.

Because a promising and suitable learning model, a decision tree, is already used for the predictors available, limited options to improve upon the work with the current data available are expected. The available improvements concern other methods that can be used to process the data. Methods that can be used are, e.g., the hypothesis tests to confirm if some of the predictors are indeed not correlated with the response, alternative cross-validation methods to compute prediction accuracy, introducing nonlinearity, more extensive tree modelling methods and unsupervised methods. Cross-validation and trees are investigated by Stellas [8], but further model extensions undoubtedly exist. Recreation of this work as presented in Appendix F, which led to similar results, gave reason to re-evaluate the used approach. The presented features' averaged nature is deemed not suitable enough for the highly variable research environment since products with vastly different layouts can still be indistinguishable in terms of the proposed feature set.

1.3.2 State-of-the-Art

Outside the Fieldlab Flexible Manufacturing, research has been performed into the effects of assistance and guidance during assembly operations. Operator guidance should aim to achieve the following five goals [4, 10]:

- raising operator awareness of relevant events;
- guiding the operator by providing instructions on time;
- monitoring the progress by collecting data;
- documenting issues;
- guarding operator status.

Operator assistance should be an added value and should, therefore, never cause annoyance by forcing or stressing the operator [11]. Furthermore, guidance should be personal due to different operator skills, operator states and operator tasks. In addition to the five goals, the assistance content (what to present), the carrier (how it should be presented) and the moment (when it should be presented, which includes automatic or not) should be considered too [11]. Assistance can also have a negative effect, if operators experience decreased difficulty during assembly, which potentially reduces motivation and alertness [4]. Specific research into the operator state-of-mind is out of the scope of this research. Negative correlations in comparison to assembly performance could however be found related to, e.g., work time or assistance level. Research into the difference between projected and displayed assistance is limited and effects of assistance technologies similar to the one available on the Omron setup on operator performance are still largely unknown [12, 13].

Display based electronic working instructions (EWI) are compared to an augmented reality (AR) based approaches particularly in E. Wilschut et. al. [14]. The two methods were split up into standard assembly steps based on single instructions and chunk instructions consisting of multiple assembly instructions. No differences were found between EWI and AR instructions, which is in contrast to later research [12]. Chunking also did not help, while it was expected that it could help in memory retrieval [15]. As the comparison of EWI and AR instructions involved a learning phase using EWI or AR and afterwards assembling without assistance, memory retrieval was expected to play a role [14]. However, it is expected that the AR method had a negative effect on the operators as they needed to adjust going from heavy assistance to none at all. The results also showed reduction of the assembly time after the learning phase, while assembly time within the learning phase already converged. This is potentially caused by the operators being held back due to the assistance, such as delays in detection. E. Wilschut et. al. [14] note that this was also observed in the work by M. Funk et. al. [16]. It was noted that assistance hinders the operators when they have already acquired the necessary skills and knowledge to assemble the product. To solve this, personalised guidance could be introduced, of which competence level, ergonomic preferences and the extent of information are examples of variables to personalise [17, 18]. Here, learning plays a significant role.

In particular, the following factors impact the learning process for operators: task complexity, structure of training programs, operators' motivation in performing tasks, and prior experience with the task [14]. Task complexity is related to the predictors introduced previously, structure of the training program is related to the different artificial intelligence modes in the Omron setup, which are introduced in section 2.3, and prior experience is related to the number of times an operator has already assembled a product. Operator motivation is yet to be explored.

Next to assistance level, operator performance can also be linked to the workload of operators. Industrial measures concerning grading of operator workload exist, which could also be correlated to performance. Operator workload is measured using the NASA Task Load Index (NASA TLX) [19], operator effort is measured using the Rating Scale of Mental Effort [20] and system usability using the System Usability Scale (SUS) [21]. While these measures are not necessarily focused on learning, they give insights into the use of assistance functions like the one of the Omron setup, which is the only way of giving instructions to the operator next to the operator screen. Research into the effects of production experience, production skills and ergonomic capability have been explored [17, 22]. Extension of the available data by tracking the operator's hands has been performed too, but limited to the use of the left hand, right hand or both hands in several phases of assembly tasks [23]. Nevertheless, it gives an introduction to using more data than the recipes and assembly performance alone.

1.4 Research Objective

The main objective of this research is to show, using assembly data, if operators assemble products in a distinguishable way, and if so, what the potential of this data is to improve operator performance via self-learning approaches of the instruction algorithm. The data is generated by operators assembling products in a high-mix, low volume environment. Improving operator performance involves decreasing assembly time and improving assembly quality by decreasing errors made.

1.4.1 Motivation

The potential business value of the results that follow from this research consists of new insights into the potential of developing learning models for high-mix, low-volume assembly setups to improve assembly instructions and, eventually, performance, which consequently increases efficiency in production. In addition, while the product type assembled on the Omron setup is simple and incomparable with commercial variants, this simplicity allows the idea and results to be easily explained, which gives it the potential to spark future research and ideas into the research domain of self-learning assembly setups.

1.4.2 Research Question

The main topics to explore during this research are operator performance, assembly of high-mix, low-volume products and self-learning potential. Due to the extensiveness of the data gathering and initial data processing, the research is focused on the potential of self-learning, which, together with the other topics, translates to the following research question:

"How much potential is there for self-learning of an assembly instruction algorithm to improve the efficiency of a flexible manufacturing assembly station in a high-mix, low-volume and operator-based production environment?"

To answer the research question, several sub-questions should be answered as summarised below:

1. Can the conclusions of the previous research by Stellas [8] be recreated with a new, more extensive data set?
2. Can a learning curve of operators be indicated?
3. Are there significant differences between the performance of different operators assembling similar products?
4. Do operators have different assembly strategy preferences?
5. Are assembly strategies correlated to performance?
6. Can custom assembly instructions be generated for operators based upon the data of other products, to improve the operators' performance?

1.5 Report Structure

This chapter introduced the parties within this research, Eindhoven University of Technology, Fieldlab Flexible Manufacturing and Omron. In addition, the motivations to support Omrons efforts into researching the possibilities of their demonstration setup are introduced. Previous work is discussed to have a foundation to build this research upon and strengthen the claims made concerning the business and research value of the main topic of this research. This topic can be summarised as research into the potential of using self-learning of assembly instruction algorithms, based on data from assembly setups with limited to no data available of products to be assembled.

The remainder of this work is structured as follows. Chapter 2 covers the knowledge needed to understand the setup used, problems faced and results presented in this research. Thereafter, Chapter 3 introduces multiple self-learning approaches for the Omron setup. The chosen self-learning technique requires a definition of similarity between products, which is introduced in Chapter 4. The gathering of data is discussed in Chapter 5. The directly derivable results are presented in Chapter 6, with included reasoning why to extend the research scope into an assembly strategy analysis in Chapter 7 and Chapter 8. Finally, conclusions and recommendations are presented in Chapter 9.

Chapter 2

Preliminaries

This chapter gives an overview of the research environment, and introduces terminologies and techniques used throughout this report. First, more details are given of the product and assembly conditions. Next, a detailed description of the product to be assembled on the Omron setup is given. Hereafter, the Omron setup itself is explored in detail. With the production environment introduced, the required prior knowledge before starting to explore opportunities for improvements in performance are introduced. These include the data that can be gathered on the setup, an introduction to the clustering of this kind of data and finally boxplot visualisation. Familiar with these preliminaries, self-learning opportunities within the Omron setup with as goal improving performance can now be introduced in Chapter 3.

2.1 Assembly Conditions

Previous and still to be introduced assembly conditions related to the Omron setup and its products are explained and elaborated upon below.

2.1.1 High-Mix and Low-Volume Products

Opposed to more traditional and well-known low-mix and high-volume manufacturing, high-mix and low-volume manufacturing refers to the production of a large variety of products in small quantities [24]. High-mix low-volume manufacturing is also referred to as make-to-order manufacturing. Depending on the product type, this results in [25]:

- Constantly changing routings. Routings determine what work is to be performed and where and how it will be performed. The large variety of product types causes the constant changing and small quantities of each of these product types.
- Quick and frequent changes of raw material on the manufacturing lines, called changeovers, due to the small quantities.
- Lack of consistency due to the large variety of the products.

The use-case presented in this research does not check all the boxes of typical high-mix low-volume manufacturing due the simplicity of the products, which is covered in section 2.2.

2.1.2 Pick-to-Light Product Assembly

Pick-to-light product assembly, or pick-to-beamer in the case of the Omron setup, refers to an assembly environment where the operator is given local visual instructions about where, which and how parts or components of the product should be picked. An example is depicted in Figure 2.1, where the number of bolts to pick is depicted on the storage bin. The placement locations of the bolts are indicated too. Assembly where the parts are parted out in storage bins is referred to as bin-picking assembly. Direct feedback to the operator is possible in position, also referred to as *in-situ*, as errors made can be indicated and corrections can be implemented in the assembly instructions itself.

2.1.3 Flexible Manufacturing

A flexible manufacturing system is a manufacturing environment where expected and unexpected changes in the production environment have been considered. The flexibility is subdivided into two categories; routing flexibility and machine flexibility. Machine flexibility allows the manufacturing process to be

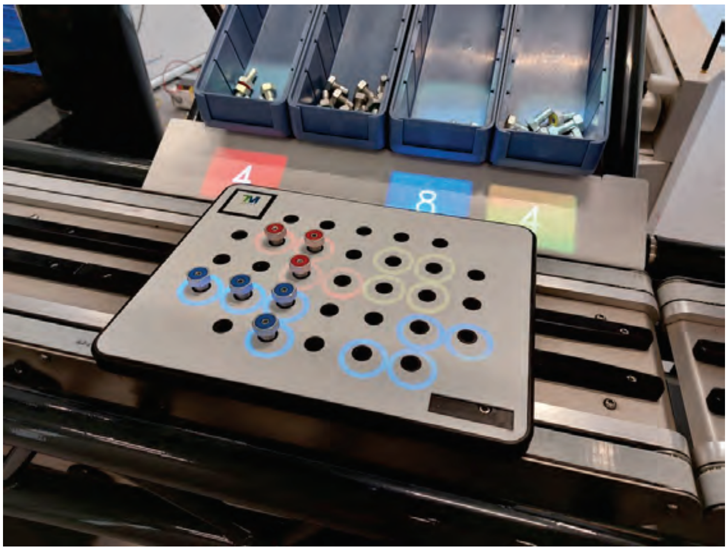


Figure 2.1: Example of pick-to-light assembly on the Omron setup

adapted to changes for new product types and changes to production orders. Routing flexibility refers to the flexibility of changing multiple setups to change the production capacity. Summarising, machine flexibility refers to the product type and routing flexibility to the product quantity [5]. This research focuses on machine flexibility as we focus on a high-mix and low-volume environment.

2.2 The Product

The products assembled on the Omron demonstration setup are based on coloured bolts that are placed and fastened inside a platform with threaded holes in a grid pattern. This grid consists out of 31 threaded holes as schematically depicted in Figure 2.2. Two schematic examples of products are depicted in Figure 2.3a and Figure 2.3b. Both products consist of four bolts of the four available colours; red, green, blue and yellow. Note the visual difference between the two examples, with the product in Figure 2.3a being generated with random bolt locations, and the product in Figure 2.3b being made manually using different patterns of bolts. Chapter 4 goes into detail about the product layout of the to be assembled products in the experiments.

In the remainder of this report, features related to the products are used. These features are partially based on the coordinates of the bolts in the products. The length axis of the coordinates, perpendicular to the operator's sight, is given by the x-axis, with the most left threaded hole equalling $x = 1$ and the most right one $x = 7$. The depth axis of the coordinates, parallel with the operator's sight, is given

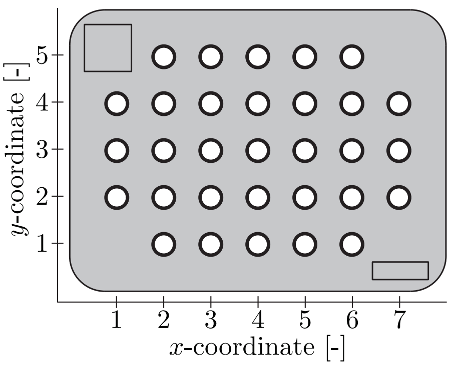
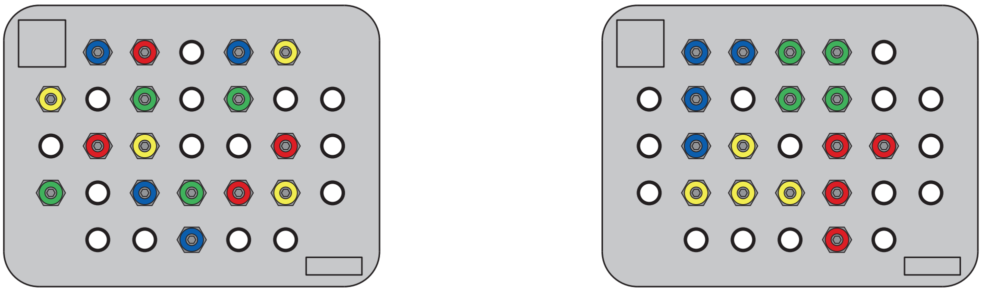


Figure 2.2: Schematic overview of the assembly platform in the Omron setup



(a) Randomly generated product (b) Custom product based on pre-defined shapes

Figure 2.3: Examples of products that can be assembled on the Omron setup

by the y-axis with the bolts closest to the operator equalling $y = 1$ and the furthest equalling $y = 5$. The axes are depicted in Figure 2.2. The combination of an integer x- and y-coordinate on a valid grid location represents a potential location where a bolt can be placed. The corners, the grid locations with coordinates $(x, y) = [(1, 1), (7, 1), (1, 5), (7, 5)]$, are missing. Every bolt has a unique bolt index for reference, starting from one and counted from left to right primarily, and from top to bottom secondarily, as given in Table D.1. The bolts can be fastened in the platform, but to save time during assembly, and due to it being a relatively constant action time-wise, a successful assembly of the product in this research only requires the operator to place the bolts in the correct position without fastening them into the platform. Collaborative robots are available for the fastening process if required.

The Omron demo setup allows for the assembly of products in a high-mix, low-volume setting due to the large number of variations possible in the assembly of a product. A product consists of at least one bolt and at most of 31 bolts in the four colours, which can be placed in arbitrary locations in the grid of holes. Therefore, every possible bolt location has five states, namely empty, yellow, green, red and blue. As there are 31 locations and since an empty product is no product, this results in $5^{31} - 1 = 4.7 \cdot 10^{21}$, i.e. almost 4.7 sextillion or 4.7 thousand billion billion possible unique products. To put this number in perspective, if all 7.8 billion people on earth would assemble a product each second for 24 hours a day each day of the year, it would still take almost 190 centuries to assemble them all. This emphasises the vast number of variants that even a seemingly simple product can have.

2.3 Overview of the Omron Setup

The Omron setup is developed as a demonstration setup for use during fairs, at Omron, and their partners. Currently, it is located at the Flexible Manufacturing Fieldlab at the Brainport Industries Campus in working condition. The setup can operate circularly as assembly and disassembly are coupled in a loop. Since it is a human-focused assembly station, the assembly itself is performed by human operators. The current operation of the setup is centred around an operator who can assemble products consisting of the introduced colour-coded bolts placed in a platform. This platform is located on a conveyor transport system for transportation through the setup. The operation routine of the setup can be changed according to the requirements of the desired research or demonstration to be executed on the setup. The following sections introduce the functions of the Omron setup.

2.3.1 Components of the Setup

All functions of the setup are depicted in Figure 2.4. The setup consists out of an:

1. operator assembly area;
2. left collaborative robot (left Cobot);
3. right collaborative robot (right Cobot);
4. beamer for the pick-to-light system;
5. depth camera for bolt placement tracking;
6. testing station camera;

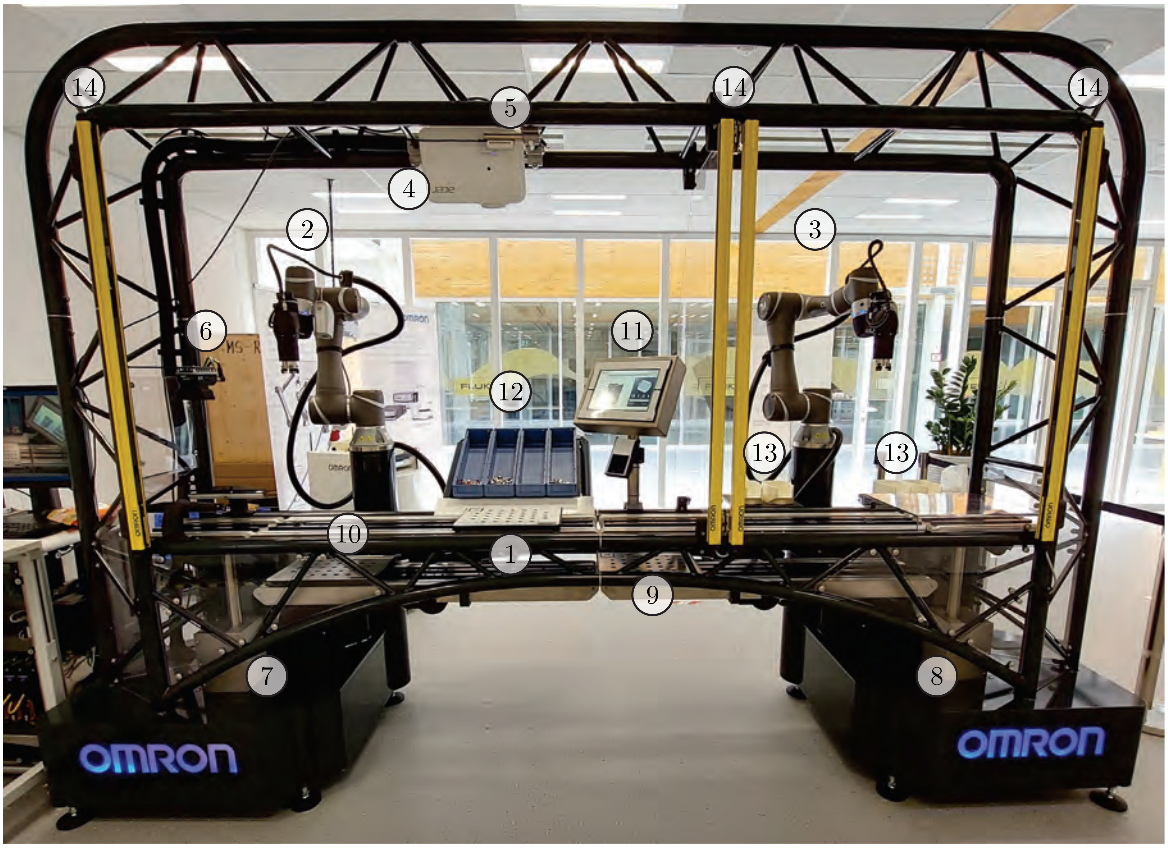


Figure 2.4: Overview image of the Omron setup

7. left elevator;
8. right elevator;
9. bottom conveyor system;
10. top conveyor system;
11. human-machine interface (HMI);
12. bolt storage location at operator;
13. bolt storage location at right collaborative robot.
14. light curtains which register when an object passes through the area of interest they are monitoring.

Additionally, the following components are part of the setup but not depicted:

- A HMI to view the data dashboard which gives an overview of the assembled products, assembly status and various other statistics.
- A HMI to view the bolt tracking neural network performance, and smart camera output.
- A moving robot which can transport bolt storage bins from Cobot to Cobot.
- A card scanner which is used for operator identification.

2.3.2 Product Assembly

The product layout with locations of the bolts is shown on the human-machine interface in front of the operator and optionally on the assembly platform itself. This is done using the overhead beamer, which depicts circles on the locations where a bolt must be placed, in the corresponding colour, see e.g., Figure 2.5. On the storage bins the beamer also depicts how many bolts of each colour have to be picked. Next to the beamer, the overhead depth camera [26] is located to track bolt and optionally hand movements of the operator. The bolts are stored by colour in storage bins behind the assembly station (AS). After placing the bolts according to the recipe, the operator can confirm that the assembly is completed, after which the platform is transported to the left using the conveyor system, where the

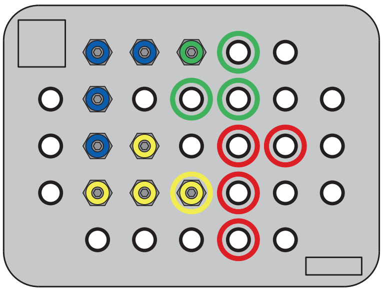


Figure 2.5: An assembly in progress on the Omron setup

locations of the bolts are checked using a second, industrial, camera [27]. After correct placement by the operator of the bolts in the platform, a collaborative robot (Cobot) with screwdriver functionality fastens the bolts. If the placement is incorrect, the platform returns to the assembly station, where the operator then needs to correct the mistake(s) using the depicted instruction. In the places where the platforms are interacted with by the operator or Cobots, the platforms lock in place, which happens only on specific places. Safety curtains are installed to force the moving parts of the setup to slow down when the operator enters their working space. A mobile robot (LD) transports bolts from the storage at the right Cobot to the main storage at the left Cobot and the operator, with the Cobots performing the transferring of the bolts between the main setup and mobile robot. Bolts are only transported when a certain threshold of bolts is reached at the right Cobot. Flow diagrams of the processes within the setup, namely the assembly platforms, the left Cobot, the right Cobot and the LD are provided in Figure 2.6. Note that only initial states are indicated and not end states. The initial states are required for start up, but as the setup is circular, there is no intended end state.

2.3.3 Assembly Instructions

Three different assembly instruction models are available on the Omron setup. The artificial intelligence assist model gives the operator the assembly steps with the lowest predicted assembly time based on gathered data on which prediction models are built. Such assembly steps are from now on called optimal assembly steps. Note that this model works per product type and per operator, and decides upon the optimal instruction set from a pre-programmed set of instructions with each one needing occurrences of it being assembled. This model is therefore not adaptable to this research's high-mix condition and desire to prevent pre-programming instructions. The artificial intelligence training mode gathers the data to create the prediction models used in the artificial intelligence assist mode. Finally, the artificial intelligence explore mode lets operators create new product type recipes.

The HMI and pick-to-light system give the actual assembly instructions for the operator. It is possible to show the product assembly instruction in the form of a complete recipe on the HMI only or both on the HMI and assembly platform. Step by step instructions are only possible to be shown on the assembly platform. These different depiction options potentially have a big influence on the accuracy and speed of an operators assembly process, as these options are assumed to affect the difficulty of the assembly.

A simulation mode is present on the setup, which allows for overruling the pick-to-light system. In practice, this means that operators can assemble products on the setup physically while data is gathered as usual, however without the Cobot, conveyor, mobile-robot and elevator functionality. The fastening and loosening functionality of the setup cannot be used in simulation mode, which is beneficial in this research as the focus is put on the placing of the bolts, and not on the fastening of the bolts. It is assumed that the fastening of bolts is a more consistent task than actually picking and placing the bolts, as here operators have to look for the correct bolt to pick and the correct location to place these bolts. As the simulation mode of the setup cannot overrule the HMI, products are assembled using the pick-to-light system, which depicts the full recipe. Assembly steps within the instructions are not used to prevent influencing assembly preferences by the operators.

When an operator is assembling a product, the schematic situation as depicted in Figure 2.5 represents a common situation. All blue bolts, represented by hexagons in the corresponding colour, are placed

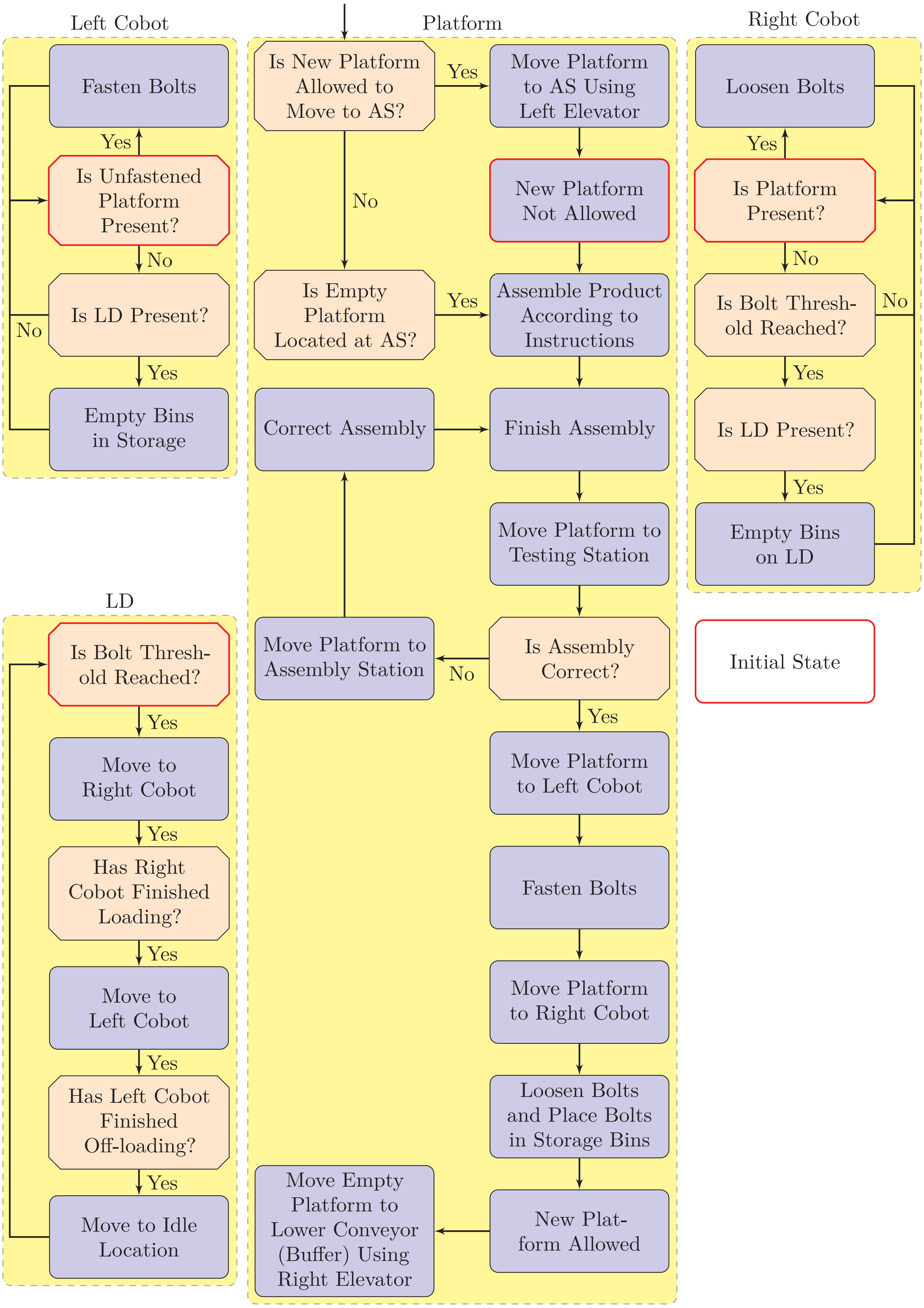


Figure 2.6: Flow diagram Omron setup, technical details concerning the activation of the conveyors, elevators and light-curtains are left out

correctly, as the lights indicated by the corresponding coloured circles around them are off. The yellow bolts are also placed correctly, but one yellow circle is shown still. This means that the neural network that uses the depth camera feed to recognise the placements of bolts, has not been able to recognise that bolt yet. Except for one green bolt, all green bolts and red bolts still have to be placed as circles are depicted by the pick-to-light system. Here, all remaining assembly steps are shown, but an arbitrary number of assembly steps and contents of these steps are possible, as long as they fall within the limits of the product recipe. If a bolt would be placed in an incorrect location, the instruction stays until the mistake is corrected.

2.3.4 Product Disassembly

After fastening the bolts during regular operation of the setup, the platform is moved to the right Cobot, which removes the bolts again and sorts the bolts by colour in separate storage bins. After a bolt threshold is reached, the right Cobot moves these bins onto the LD, which transports the bins to the operator storage bins where the left Cobot empties the bins. This loop makes it possible to let the system run continuously without intervention, making it circular. Bolt loosening and fastening have priority over transferring bolts on and from the moving robot. During data gathering, loosening of the bolts as well as placing the bolts back into their storage bins is performed manually by the operators due to the use of the simulation mode.

2.4 Data Gathered

The Omron setup tracks an extensive list of data, which is presented in Appendix D.1. For now, only the relevant data for this research, namely the data gathered by the pick-to-light system, is introduced. This data consists out of every placed bolt's location on the platform grid, colour, and time of placement.

As introduced, the bolt placement is tracked by a neural network using a vision system consisting of a colour depth camera [26] above the assembly platform. In addition to this data, the corresponding product identification number (ID), the product type ID, referring to the layout of the product, and operator ID are also known. This allows to check if the bolts are correctly placed according to the product that was supposed to be assembled. In data post-processing steps the assembly data can be coupled with the product ID, product type ID and operator ID. The start of the assembly based on turning on the pick-to-light instructions is known, but since it is unknown what happens between that time and the actual assembly start, the assembly time is defined as the time between the first and last placed bolt of a product.

A basic hand tracking system is currently available, but the data is not logged. A neural network is trained to recognise blue gloves worn by an operator. Both hands can be tracked based on each hand's central location, which gives access to additional data. Therefore, next to improving the current data processing, there is potential to extend the available data, which is discussed in Chapter 3.

2.5 Visualising Performance Using Boxplots

A boxplot is a commonly used method of visualising measurement data and comparing if different subsets have significant difference in value. A subset is defined by the same value of a certain feature present in the complete data set. A boxplot depicts the subsets:

- Median, also called second quartile (Q2). It represents the middle value of the samples belonging to the same common feature value. This is not the mean value.
- The first quartile (Q1) and third quartile (Q3) respectively depict the middle value between the smallest sample and the median, and the middle value between the median and the biggest sample.
- The interquartile range (IQR) represents the data points between the first and third quartile.
- The bottom and top horizontal lines respectively represent the values Q1 percentile $-$ 1.5 IQR and Q3 percentile $+$ 1.5 IQR. Note that the lines have to lie on the nearest existing data point not further away from the median value than the values respectively represent. The values are respectively denoted by the minimum and maximum value.

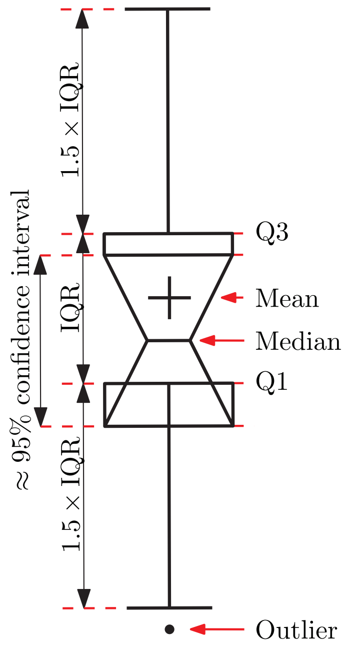


Figure 2.7: Overview of a boxplot [28]

- While mathematically redundant, the vertical lines are called whiskers.
- Data points with values less than the minimum or greater than the maximum represent outliers. For readability of the figures, these are not depicted in the figures that present results of Chapter 6.
- The notches represent the confidence interval of the median. This confidence interval is based on Gaussian-based asymptotic approximation [28]. It represents a 95% confidence level around the median based on the IQR and number of samples. Note that when the upper or lower confidence interval extends beyond respectively the third or first quartile, the notch is inverted. In comparison to literature, the visualisation of the notches is slightly adapted for the benefit of readability [28]. To this end, horizontal lines at the endpoints of the notches are added.

2.6 Summary

This chapter presented the preliminaries necessary to follow the research topics presented in this report. The assembly conditions are introduced, which are high-mix, low-volume, flexible, and pick-to-light assembled. Furthermore, the product itself is presented, an overview is given of the assembly setup and the data that is gathered. Finally, the boxplot visualisation method that is often used in this thesis, is introduced. The next chapter covers the self-learning potential and self-learning opportunities the Omron setup has.

Chapter 3

Self-Learning Opportunities

Now familiar with the Omron demonstration setup and the high-mix, low-volume product type, this chapter presents an opportunity for operator performance improvement in the form of self-learning. First the possible data sources are introduced by exploring the potential of bolt placement tracking and an extension to hand tracking to emphasise the limitations of solely using bolt placement tracking. Because self-learning algorithms in a high-mix, low-volume production environment do not have access to substantial data samples of specific products, product similarity is introduced to explore the potential of grouping unique products to increase sample size. Having introduced the necessity of a similarity measure, Chapter 4 introduces how such a measure can be defined.

3.1 Bolt Placement Tracking

The Omron demonstration setup tracks the placement time, location and colour of bolts using a camera above the assembly platform, via a neural network algorithm approach. When combining this data with the product layout, four-dimensional data is available consisting of the x-coordinate, y-coordinate, colour and placement time of all placed bolts. The placement times are zeroed to the placement time of the first bolt. This data can be combined with the other available data such as operator ID and absolute starting time to have a more complete picture of the assembly conditions. By making use of the combined data, features like how many products a certain operator has assembled in total or per type can be derived. There are self-learning opportunities within this data related to the patterns operators use. For example, in what order do operators work, and can these be correlated to product features? Are there patterns within the assembly pattern, for example jumps in time that indicate when the operator picks bolts or searches for the correct placement location? Also, does performance change over time?

A limitation of bolt placement tracking is however that only one part of the assembling process of products is tracked, namely the actual placement of bolts. Much information is missing here, like picking of bolts and assembly trajectories, which could be extracted with the use of hand tracking. Hand tracking would allow gathering assembly trajectories that contain information before, between and after the placement of bolts.

3.2 Hand Tracking

Currently, Omron has a basic hand tracking algorithm working on the Omron setup, but no data is logged. This tracking is in the stage that two *blobs* are available for every hand. These blobs are logged in three-dimensional space using the same depth camera [26] that tracks the bolts. Currently, the operators are required to wear blue gloves for hand tracking. Tracking the hands makes it possible to log motion paths of both hands. By combining this data with the bolt placement recognition neural network, a motion track of each hand can be synced with the placements of bolts based on the logged time. Basic movement sequences, called *motion primitives*, could be extracted from these trajectories.

3.2.1 An Introduction to Motion Primitives

Motion primitives form the basic patterns of movement sequences by which every movement can be made [29]. A well-known definition of motion primitives is *Gilbreth's Therblig*, which describes all possible basic movements within an assembly process [30]. In total, there are 18 Therblig motion primitive definitions. If all sub-motion sequences of an assembly can be labelled with Therbligs, a lot of information is gained concerning efficiency of an operator. Time spent in certain Therbligs can be compared to other operators, with some Therbligs being by definition bad for performance (e.g., *Avoidable Delay*), and others being needed to complete an assembly (e.g., *Transport Loaded*).

Certain Therbligs indicate motion sequences of the assembly that can influence the quality of assembly assistance. Extending the blob information to knowing which hand is the left one and which hand is the right one, allows logging data for the right and left hand separately. Furthermore, statistical information about how much time spent in each motion primitive and the trajectories itself could be potentially interesting for operator recognition, which is an open field of interest for Omron. Syncing the trajectories with the placement of bolts also gains information about the number of bolts picked at a time, since picking inormation can be derived from the merged data. For completeness, all Therbligs are given in Appendix B.

3.2.2 Potential of Motion Primitives

The already available data of the bolt placements are cut-off points for time segments, and position segments that cover one instance of a motion primitive, as the placement of a bolt indicates both the end and beginning of a new motion primitive, in this case *Release Load*. Classification based supervised learning, correlation-based unsupervised learning and neural networks have the potential of recognising motion primitives, with as input either split segments of the position over time or trajectory only. More information about learning models can be found in the book *An Introduction to Statistical Learning* by G. James et. al. [9]. The benefit of position over time versus trajectory only would for example be knowing when the operator is standing still.

Much information can already be derived from the locations a hand visits without making use of learning algorithms. If a hand hovers over a certain bin with a certain colour, the operator is probably picking bolts if he or she thereafter also places that kind of bolts. If an operator hovers over a bin of the colour he just placed and then visits another bin, he or she is probably returning bolts. If an operator places multiple bolts without visiting a bin, he or she picked multiple bolts. Like this, a simplified identification of the motion sequences using motion primitives can already be created using a subset of Therbligs, e.g., *Search*, *Transport Loaded*, *Transport Loaded*, *Assembly*, and *Release Load*. Additionally, information is now known about the number of bolts picked, transported, placed and returned to bins. This additional information is not known by solely using the bolt-placement data, so this simplified identification already gives useful additional information.

3.2.3 Uncertainties Concerning Motion Primitives

Next to the potential of motion primitives, some risks and uncertainties should be noted. Currently, the resolution and sample rate of the hand tracking is unknown, which makes it difficult to assess the potential of actually being able to gather useful data to identify motion primitives. Also, the data will be limited to the hands only in the short term, limiting the information to identify the number of motion primitives. If implemented, it should be investigated if the camera integrated into the end-effector of the Cobot or a new camera gives additional useful information, if the top view camera proves too limiting.

3.2.4 Identification of Motion Primitives State-of-Practice

An existing implementation of Therblig identification is based on guiding robotic arms [29]. Position data is recorded using the motor encoders, and velocities are estimated using an extended Kalman filter. In the Omron setup, only position data of the hand is available, which can be compared to the end-effector's position. The benefits of robotic arms are that gripper information can be obtained by the current delivered to it and that force and torque data is available. This is information which is not available using a camera. Also, orientations are easily derived on the robot, while hand orientation derivation using a camera is less straightforward. Therblig identification was implemented using a $K-1$ classes support vector machine that sequentially decides on class 1 versus $2, 3, \ldots, K-1, K$, between class 2 and $3, 4, \ldots, K-1, K$ et cetera. The class with the highest occurrence is chosen. Every data sample is given a class, so no trajectory segmentation is performed. A subset of five Therbligs is used.

Research into learning pick-and-place tasks for robots also applied Therblig identification by using hand data [31]. The hand position was recognised from the camera feed based on several filters and edge detection. Closed and opened hands could also be distinguished [32]. The Therbligs that can be recognised are *Transport Empty*, *Grasp*, *Transport Empty* and *Release*. Moving or not moving decides between transporting or not, and the detection of fingers or not decides on the hand grasping or not. Motion data can be segmented and classified in Therbligs by defining keyframes in which motion is happening [33]. Feature points are recognised, which in the case of the Omron demo machine, are the bolts. A support vector machine is again used for the classification.

Research also exists into learning of the motion primitives itself, without the use of a motion primitive library like Therblig's library [34, 35]. This method uses probabilistic segmentation. Since this research's use-case is quite specific and as suitable classes are proposed in he form of Therbligs [30], creating a custom library is considered out of the scope of this research. Segmentation of the data could however be interesting, which is further discussed in the work of Lemme et. al. [36].

The current data gathered from the Omron setup already gives enough learning potential, so it is decided to leave the possible application of motion primitives to the setup as a recommendation for future research. However, the principle is still introduced due the high potential of learning and tracking more information of operators' performance during assemblies which can be used in extensions of the research presented in this report.

Using the bolt placement data alone is not sufficient to make a learning algorithm, due the introduced problem of lacking data of to be assembled products. To define which data is useful for a new product which needs to be assembled, a similarity measure of both products and assemblies needs to be defined that can grade data of non identical products on usefulness in a learning algorithm. Data of non identical products could still be useful due to overlapping, to be defined, features.

3.3 Product Similarity Measure

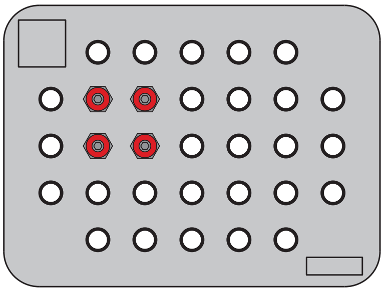
If a measure of similarity is available for the different product types, assembly sequences of operators can be compared for similar products. If patterns can be found, the data of the most similar products that are already assembled can be used for the new, still to be assembled, product. Similarity is challenging to define, but within this research similarity is measured and graded in the context of assembly sequence similarity; if two products are assembled in the same way, the products are similar. Therefore, a subsequent step is to define what it means to assemble a product in the same way. The problem is that product similarity should be known before the assembly takes place, as optimised assembly instructions cannot use previous data otherwise. This previous data is assumed to be required to originate from similar products, which explains the necessity of knowing which products are similar.

3.3.1 Similarity Measure Definition

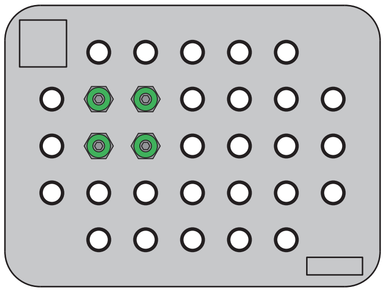
To define a similarity measure, simply comparing corresponding absolute locations on the platform grid is insufficient, as this does not represent similarity for humans. This is extensively described by the Gestalt principles, which go into detail on why humans group objects based on similarity and proximity [37]. Both these Gestalt principles are present in the setups bolt based products, namely by respectively colour and position.

When comparing two products, the shapes the bolts make and how many translations, rotations and mirror operations of these shapes are needed to get from one product to the other product are expected to be relevant. This because it is expected that humans use groups of bolts, or shapes, to compare products and decide on similarity. The uniformity of the colours is also expected to influence similarity as a uniformly coloured product is expected not to be equal to a product with the same bolt locations in many different colours. However, a single bolt shape, or *cluster*, in two products with different colours, can be very similar. Multiple shapes of bolts that combined form bigger clusters of bolts also have potential of a higher similarity if products have the same cluster, as these can potentially be build in one go.

Consider the following example: two products with the same bolt locations but different colours can be as similar on their own, as can be seen in the comparison of products $P_{3.1a}$ and $P_{3.1b}$ in respectively

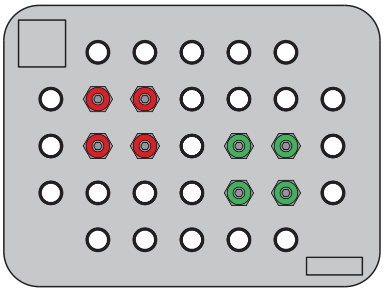


(a) Product $P_{3.1a}$

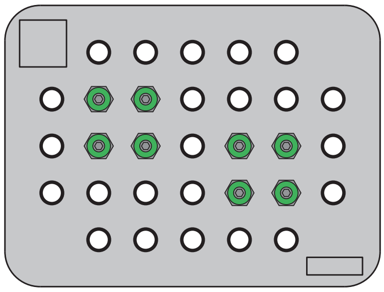


(b) Product $P_{3.1b}$

Figure 3.1: Product examples to demonstrate the difficulty of defining similarity. While the two 2×2 bolt clusters have different colours, the author would tackle this task the same disregarding different locations of the storage bins of the different coloured screws



(a) Product $P_{3.2a}$



(b) Product $P_{3.2b}$

Figure 3.2: Product examples to demonstrate the difficulty of defining similarity. Again the two 2×2 shapes, also depicted in Figure 3.1, have different colours, but as there is also a second bolt cluster present in both products, these products are not as similar as products $P_{3.1a}$ and $P_{3.1b}$ are

Figure 3.1a and Figure 3.1b. These products are very similar, as there is no reason to assemble them differently, with the exception of the bin locations of the colours being different with respect to the bolts to be placed, which could influence the order of placement of the bolts. For now, ignore the influence of the bin locations. If other bolts or bolt clusters are present, this similarity can differ, as can be seen in the comparison of products $P_{3.2a}$ and $P_{3.2b}$ in respectively Figure 3.2a and Figure 3.2b. While the same bolts are added, the similarity changes are unequal between products $P_{3.1a}$ and $P_{3.2a}$, versus products $P_{3.1b}$ and $P_{3.2b}$.

The features of the proposed similarity features, the translation, rotations and mirror operations of (clusters of) bolts, can be described mathematically. However, they all could have different weighting factors, which also could be dependent on the other feature values when compared to a manual grading of product similarity. Therefore, an algorithm based on these features certainly would have to be extensively tuned, and would be operator dependent and potentially even programmer dependent as the weight allocation is a subjective process.

3.3.2 From Product Similarity to Assembly Instruction

When similar products can be identified, the way these products are assembled can be looked into to see if it can be predicted what the fastest error-free assembly method is of the new product. This method can then be translated into an assembly instruction for the operator. As it will be challenging to gather assembly data in big amounts, some overlap of the product types between operators should be present. Like this, data of multiple assemblies of the same product type is available to actually know the fastest way of assembling, as optimal operator performance cannot be guaranteed during the entire operator's work shift. Having a substantial sample size is required to filter relatively bad performance, as grading the similarity of products based on assembly data of bad performance breaks the proposed approach of learning from similar product assemblies. As the product assemblies are performed solely for the purpose of this research, it is possible to incorporate repetition.

3.4 Other Learning Opportunities

Another self-learning opportunity, related to features of the Omron setup, is recognising operators, as this nullifies the need to log in for personal assembly assistance, see, e.g. the work of R.J.A. Cockx [8]. On the hardware side, adaptive bolt fastening would also be beneficial, as currently, due to wear of both the bolt and platform thread, sometimes bolt fastening or loosening fails. A full list of the learning potential concerning the setup is presented in Appendix A.

3.5 Conclusion

Trajectory data of the hands in combination with (a simplified version of) the proposed motion primitives proposed gives additional information about the picking and searching sequences within the complete assembly sequence of an operator. However, due to the already big potential and challenges of only the bolt placement data, this research focuses on analysing the performance of bolt placement of different product types by different operators, with an underlying comparison of the products based on their similarity. This similarity is further explored in Chapter 4 and is based on to be defined product features.

Chapter 4

Product Similarity Measures

In this chapter, various product similarity measures are explored. Chapter 3 introduced why such a measure is important: to define a grouping of assembly data with a suitable sample occurrence that can be used to create the optimal assembly instruction for an operator, with a greater sample size in comparison to only using the exact product to be assembled with no guarantee of being assembled before. Chapter 3 also touched upon the difficulty of defining a similarity measure, which is extended upon in this chapter by walking through possible similarity measures. Benefits and downsides of the considered measures are discussed, and improved upon in the consecutive suggestions. Finally, an alternative method of defining similarity in the products is introduced, which is weighted against the similarity measures. At the end of this chapter, a suitable similarity measure is chosen to allow comparing assembly performance with suitable sample sizes based upon the experiments defined in Chapter 5.

4.1 Summary Based Similarity

The products and their assembly instructions can be described using certain features which can have a limited subset of values, which allows to describe the products and assembly instructions in a summarising way. Like this, the necessary overlap between the much larger number of possible products and even larger number of possible assembly instructions with no requirement of repeating occurrences. These features are:

- bolt column density (fill density per x-coordinate);
- bolt row density (fill density per y-coordinate);
- number of bolts;
- number of bolt clusters;
- cluster density (number of bolts belonging to a cluster; divided by the number of bolts);
- ratios of colour distributions.

Similar features can be defined for the assembly instructions, as summarised in section 1.3.1. These features are introduced in the work by Stellas [8] and recreated in Appendix F with the new data set and new, to be introduced, product types. No new results were gained based on this new data set.

Unfortunately, due to the nature of these features being an averaged value, there is no guarantee that products with similar feature values are actually similar. Adding a standard deviation value could help, but increases the potential number of combinations of feature values significantly. Furthermore, due to the limited number of certain feature values, the values itself and therefore the standard deviations too, have limited values which gives a distorted view. The desired similarity measure needs to incorporate the layout of the products. A vector notation of the product incorporates all information necessary to be a unique representation of the product. A measure to compare vector notations of different products has been found in *edit distances*, which define the number of edits, or operations, in a string that need to be made to go to another string [38].

4.2 Edit Distance Based Similarity

The possible operations that can be used by different edit distance definitions are deletion, insertion, substitution and transposition (swapping) [38]. Edit distances are often used in spelling correction applications, as words with a close distance to an unknown word may be the intended word by the writer.

This measure of closeness of words has potential of being translatable to a to be defined notation of the products, where the edit distance would define a measure of similarity of the products.

Different types of edit distances allow different subsets of the complete set of possible operations listed above. The first step in creating a measure is to define which string notation should represent the product layout. As an initial proof-of-concept, the notation is based on a string of 31 characters, based on the bolt indices, $B_i \in [1, 2, \dots, 31]$ defined primarily left to right and secondarily top to bottom. For a complete overview of the index numbers, refer to Table D.1. If a bolt index location is occopied by a bolt, this is indicated with the colour of the bolt, with r for red, g for green, b for blue and y for yellow.

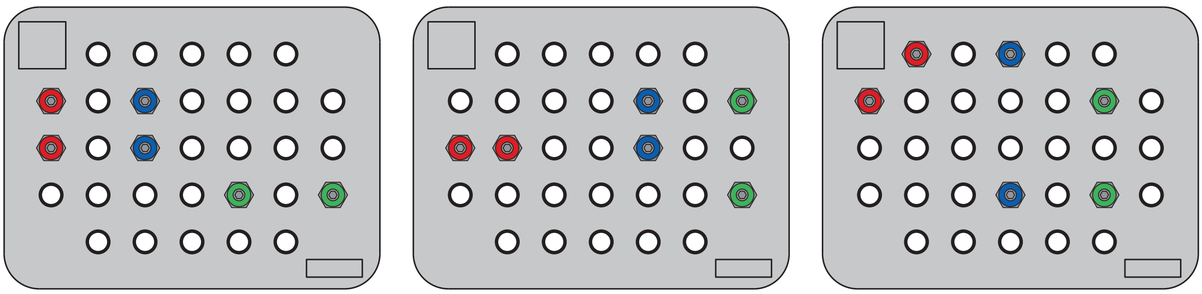
Three example products, products $P_{4.1a}$, $P_{4.1b}$ and $P_{4.1c}$, as depicted in Figure 4.1, are defined. The, shortened, bolt indices vector $\vec{B}_i$ and bolt strings $\vec{R}_p$ of the corresponding products p, based on above notation definition above, are given by

$$\begin{array}{rcccccccccccl} \vec{B}_i = & [& 1, & 2, & 3, & 4, & 5, & 6, & 7, & 8, & \dots, & 31 &], \\ \vec{R}_{4.1a} = & [& r, & r, & -, & -, & -, & -, & -, & -, & \dots, & g &], \\ \vec{R}_{4.1b} = & [& -, & r, & -, & -, & -, & r, & -, & -, & \dots, & g &], \\ \vec{R}_{4.1c} = & [& r, & -, & -, & r, & -, & -, & -, & -, & \dots, & - &]. \end{array}$$

Comparing the products based on transpositions alone, the red bolts always have a change of one bolt compared to product $P_{4.1a}$, as only one transposition needs to be performed. So, based on the edit distances of the product strings of the red bolts only, products $P_{4.1b}$ and $P_{4.1c}$ are as similar to product $P_{4.1a}$, while product $P_{4.1b}$ can be considered as more similar to product $P_{4.1a}$ than product $P_{4.1c}$. For the blue bolts, both bolt positions change, so all would be as similar, while product $P_{4.1a}$ and product $P_{4.1b}$ seems a lot more similar than product $P_{4.1c}$ to either of the others. For the green bolts, product $P_{4.1a}$ and product $P_{4.1b}$ can be considered more similar due to their common bolt than product $P_{4.1b}$ to product $P_{4.1c}$, while these can be considered more similar.

Again, it can be said that relative distances and groupings are lacking in the measures. Furthermore, the different operations should be weighted relative to each other as, for example, transposition by definition means that the correct bolts are available to transform one product in another, while substitution means bolts have to be added or removed. Also, the number of operations that need to be performed depend on the difference in number of bolts used, which would effect the edit distance and therefore similarity.

As an extension, a distance measure is added, by using the x- and y-coordinates of the bolts. The bolt indices lack this information, so the relative distances between the bolts can now be taken into account. This edit distance is defined as the number of steps in x- and y-direction the bolts have to be moved to get to the other layout. Note that equally sized products with an equal number of every colour are assumed. Table 4.1 summarises the edit distances of the three colours in the three examples of Figure 4.1, with the lower triangular entries depicting the index based edit distances, and the upper triangular entries the coordinate based edit distances. Again, the index based edit distance is based upon transposition. Considering the red bolts only, products $P_{4.1a}$ and $P_{4.1b}$ are more similar intuitively, but the index based method indicates that products $P_{4.1a}$ and $P_{4.1c}$ are as similar, the coordinate based edit distance therefore



(a) Product $P_{4.1a}$ (b) Product $P_{4.1b}$ (c) Product $P_{4.1c}$

Figure 4.1: Examples of the limitations of edit distance based on bolt index number and bolt coordinates

Table 4.1: Comparison of index based, in colour ■, and coordinate based, in colour ■, edit distances

	red bolts			blue bolts			green bolts		
product	$\mathbf{P_{4.1a}}$	$\mathbf{P_{4.1b}}$	$\mathbf{P_{4.1c}}$	$\mathbf{P_{4.1a}}$	$\mathbf{P_{4.1b}}$	$\mathbf{P_{4.1c}}$	$\mathbf{P_{4.1a}}$	$\mathbf{P_{4.1b}}$	$\mathbf{P_{4.1c}}$
$\mathbf{P_{4.1a}}$	-	2	3	-	4	4	-	4	5
$\mathbf{P_{4.1b}}$	1	-	3	2	-	2	1	-	2
$\mathbf{P_{4.1c}}$	1	2	-	2	2	-	2	2	-

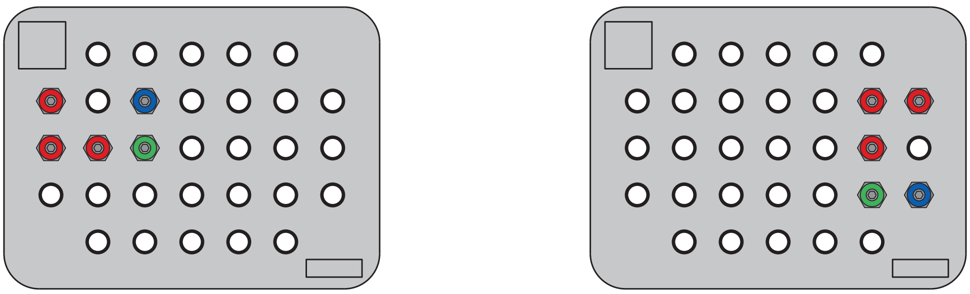
aligns better with intuition. Considering the blue bolts only, all products are as similar index based, but products $P_{4.1a}$ and $P_{4.1b}$ are the most similar intuitively, coordinate based says products $P_{4.1b}$ and $P_{4.1c}$ are more similar. Finally, considering the green bolts only, products $P_{4.1a}$ and $P_{4.1b}$ are more similar index based, while products $P_{4.1b}$ and $P_{4.1c}$ are more similar intuitively, which is in agreement with coordinate based edit distance.

It is clear that the edit distance is not suitable to define similarity between products, as due to the way humans group sets of objects, directly comparing locations of single bolts, which the edit distance does, is not suitable. Comparing a bolt on index one, with one on index two, three, four or five have very different relational properties. A group of bolts translated globally over the grid without local changes, is still very similar, while local versus global transformations are not taken into account with edit distances as defined in this section. Therefore, the next step is to look into these groupings of bolts, previously introduced as bolt *clusters*.

4.3 Bolt Cluster Based Similarity

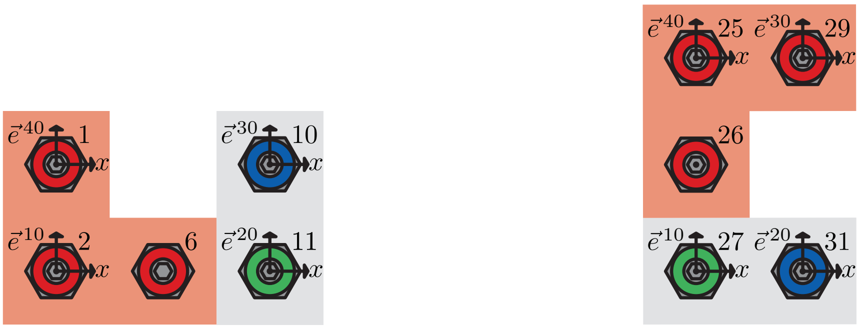
Consider products $P_{4.2a}$ and $P_{4.2b}$, depicted in Figure 4.2. If the edit distance based similarity would be applied on the comparison of these two products, they would not be similar. However, the products are actually quite similar if the bolts are seen as one or multiple clusters which are rotated and translated versions relative to each other. To define a similarity measure between possible clusters, these clusters have to be identified first. A cluster is defined as two or more adjacent bolts in x- or y-direction. Not in diagonal direction, but this could be added based on experimental results if operators see diagonally connected bolts as clusters too. An additional distinction can be made between clusters existing out of the same colour or out of arbitrary colours. The neighbours of every bolt with location (x, y) are checked by checking the grid positions on locations $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$, and if any are occupied by a bolt, it is added to the cluster. Bolts that have been added to a cluster are not checked again as bolts cannot be part of multiple clusters, following the introduced definition of a cluster.

Within a cluster, four origins of local coordinate systems are introduced: $\vec{e}^{10}, \vec{e}^{20}, \vec{e}^{30}$ and $\vec{e}^{40}$, as depicted in Figure 4.3 for both products consisting out of clusters of bolts with arbitrary colours. The reason why four local origins are defined and how they are defined, will become clear later in this section. In $\vec{e}^{mn}$, $m \in [1, 2, 3, 4]$ indicates the local coordinate system number and $n \in [0, 1, 2, 3]$ the rotation of the local coordinate system in $90n°$. The numbers in the top right corner of every bolt in the visualisation depict



(a) Product $P_{4.2a}$ (b) Product $P_{4.2b}$

Figure 4.2: Product examples showing of bolt clusters



(a) Local coordinate systems product $P_{4.2a}$ (b) Local coordinate systems product $P_{4.2b}$

Figure 4.3: Definition of the local coordinate systems of bolt clusters

the bolt index number. Connected squares in the same colour depict clusters of bolts in the same colour, with the exception of the colour grey which indicates clusters of bolts in different colours. Together they form a bigger cluster too in the case of depicted examples in Figure 4.3. The positive x-direction of a local coordinate system is indicated with an x and arrow, with the other arrow defining the positive y-direction. The order of the local coordinate systems is based primarily on lowest y-coordinate and secondarily on going from lowest to highest x-coordinate. So starting in the bottom-left corner. The bolt locations of product $P_{4.2a}$ in the local coordinate systems are given by

$$\begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3a}\vec{e}^{10}} = \begin{bmatrix} 0 & 0 & 1 & 2 & 2 \\ 1 & 0 & 0 & 1 & 0 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 1 & 2 & 6 & 10 & 11 \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3a}\vec{e}^{20}} = \begin{bmatrix} -2 & -2 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 1 & 2 & 6 & 10 & 11 \end{bmatrix},$$

$$\begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3a}\vec{e}^{30}} = \begin{bmatrix} -2 & -2 & -1 & 0 & 0 \\ 0 & -1 & -1 & 0 & -1 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 1 & 2 & 6 & 10 & 11 \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3a}\vec{e}^{40}} = \begin{bmatrix} 0 & 0 & 1 & 2 & 2 \\ 0 & -1 & -1 & 0 & -1 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 1 & 2 & 6 & 10 & 11 \end{bmatrix},$$

with x the bolt's x-coordinate, y the bolt's y-coordinate, c the bolt's colour and B_i the bolt's bolt index. Bolts are sorted in increasing bolt index value. The bolt coordinates of product $P_{4.2b}$ in the local coordinate systems of product 4.3b are given by

$$\begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3b}\vec{e}^{10}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 & 0 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 25 & 26 & 27 & 29 & 31 \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3b}\vec{e}^{20}} = \begin{bmatrix} -1 & -1 & -1 & 0 & 0 \\ 2 & 1 & 0 & 2 & 0 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 25 & 26 & 27 & 29 & 31 \end{bmatrix},$$

$$\begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3b}\vec{e}^{30}} = \begin{bmatrix} -1 & -1 & -1 & 0 & 0 \\ 0 & -1 & -2 & 0 & -2 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 25 & 26 & 27 & 29 & 31 \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ c \\ B_i \end{bmatrix}_{{}^{4.3b}\vec{e}^{40}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & -1 & -2 & 0 & -2 \\ \mathrm{r} & \mathrm{r} & \mathrm{r} & \mathrm{b} & \mathrm{g} \\ 25 & 26 & 27 & 29 & 31 \end{bmatrix}.$$

By comparing the local coordinate systems of the clusters of two products, a match is found when both the bolts of both clusters match in their relative positioning, which means that globally, the clusters may be translated and rotated versions of each other. To find a match, all cluster combinations between two products are sequentially turned in every sub-coordinate system. The rotated systems of $\vec{e}^{10}$ are given by $\vec{e}^{11}, \vec{e}^{12}$ and $\vec{e}^{13}$, which are respectively rotated 90°, 180° and 270° compared to $\vec{e}^{10}$, with positive rotational direction anti-clockwise, as defined by the main Cartesian coordinate system. Coordinates can be defined in all, rotated, local coordinate systems, as given by

$$\begin{bmatrix} x & y & c & B_i \end{bmatrix}^\top \vec{e}^{mn} = A(90n) \begin{bmatrix} x & y & c & B_i \end{bmatrix}^\top \vec{e}^{m0} \text{ with } m \in [1,2,3,4] \text{ and } n \in [0,1,2,3], \quad (4.1)$$

with

$$A(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{4.2}$$

The local bolt coordinates for the cluster of product $P_{4.2b}$ in these coordinate systems with origin one are given by

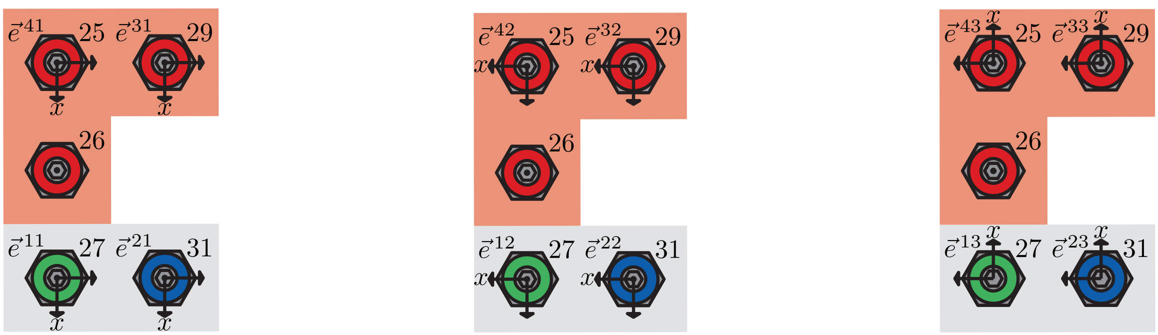
$$\begin{bmatrix} x \\ y \\ c \end{bmatrix}^{4.3b\vec{e}^{11}} = \begin{bmatrix} \text{-2} & \text{-1} & 0 & \text{-2} & 0 \\ 0 & 0 & 0 & 1 & 1 \\ \text{r} & \text{r} & \text{r} & \text{b} & \text{g} \end{bmatrix}, \quad \begin{bmatrix} x \\ y \\ c \end{bmatrix}^{4.3b\vec{e}^{12}} = \begin{bmatrix} 0 & 0 & 0 & \text{-1} & \text{-1} \\ \text{-2} & \text{-1} & 0 & \text{-2} & 0 \\ \text{r} & \text{r} & \text{r} & \text{b} & \text{g} \end{bmatrix},$$

$$\begin{bmatrix} x \\ y \\ c \end{bmatrix}^{4.3b\vec{e}^{13}} = \begin{bmatrix} 2 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & \text{-1} & \text{-1} \\ \text{r} & \text{r} & \text{r} & \text{b} & \text{g} \end{bmatrix}.$$

These coordinate systems and the rotated variants are depicted in Figure 4.4. The x-coordinates, y-coordinates and colours of $\vec{e}^{20}$ of product 4.3a and $\vec{e}^{11}$ of product 4.3b, also referred to as respectively ${}^{4.3a}\vec{e}^{20}$ and ${}^{4.3b}\vec{e}^{11}$, are equal, meaning that product 4.3b is rotated with respect to product 4.3a with 90° anti-clockwise. Note that the order within the matrices may differ, as this order is arbitrary. Taking the absolute coordinates of both the sub-coordinate systems, respectively equal to $(4, 3)$ and $(6, 2)$ the *Manhattan distance* can be defined. The Manhattan distance measures distances in the separate distances traversed in x- and y-direction [39]. With a $|\Delta x| = 2$ and a $|\Delta y| = 2$, this gives a Manhattan distance of four.

Multiple local coordinate systems have to be introduced for every bolt cluster, to make sure matching clusters have at least one matching local coordinate system in the same matching origin, as these origins are not known beforehand. Note that for eventual implementation of this approach, an extensive check should performing if at least one of these four local coordinates will always match with a rotated version of the same cluster with a local origin allocation algorithm based on the four corners of a cluster (top-left, top-right, bottom-right, bottom-left).

In the given example, all origins match, so if all origins and rotations are checked, four matches would be found. The algorithm can be extended to partial matching to allow small mismatches in the local coordinates to allow for finding similar, but unequal, clusters in products. The cluster approach however only covers the finding of clusters and (partial) matching of these clusters. How to grade similarity based on (partial) matching of these clusters remains an open question, with many variables unknown like the effect of cluster size, number of clusters and the exact definition of a cluster in the viewpoint of a human. As the relative positioning between bolts and comparing the positioning between clusters, touches on the research field of *graph theory*, this field is explored next in section 4.4.



(a) Rotated 90° (b) Rotated 180° (c) Rotated 270°

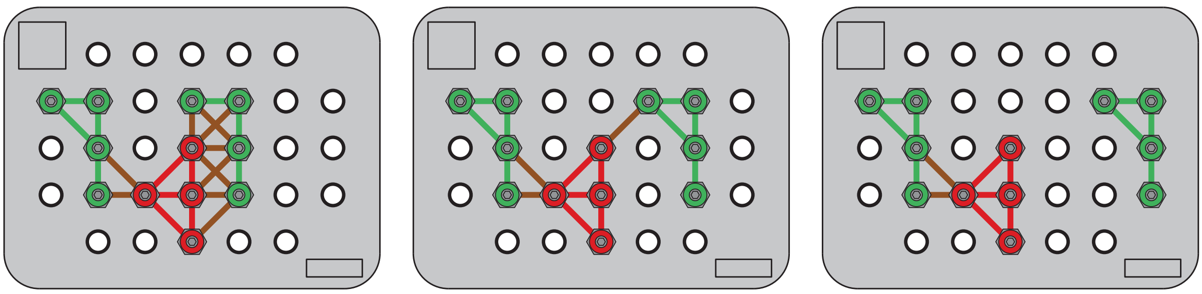
Figure 4.4: Rotations of bolt sets. x-direction is indicated, and arrows point into positive direction

4.4 Graph Based Similarity

The way similarity between products is sought after, based on finding matching groups of bolts in the compared products, can be translated to graphs. Bolts are represented by nodes and their adjacency to other bolts is represented by edges. The edge colours represent the colours of the two nodes that they connect. These edges can be introduced in x- and y-direction, but also in diagonal direction. The edges may extend to bolts only one grid point away, to the neighbouring bolts of maximum $\sqrt{2}$ grid points away, or even further.

The goal is to find similarity between the graphs, based upon graph similarity. Edges can be created in multiple ways, of which a neighbour detection variant is implemented. Every set of nodes within a distance of $\sqrt{2}$ grid points of each other, are connected via edges. Figure 4.5 depicts the graph notation of products $P_{4.5a}$, $P_{4.5b}$ and $P_{4.5c}$. To check if products are equal or contain equal parts using graphs, no degrees of freedom may be left in the graph that could allow a graph with a fixed number of nodes, edge lengths and colours to match with two non-unique (parts of) products. Degrees of freedom are defined as it being possible to re-position at least one node (bolt) in a way that the graphs stays equal, meaning no change in nodes, in edge lengths and in which nodes the edges connect. It is easily explained by the following example. Assume that the nodes are not fixed to the platform, and only connected to each other by the edges that can freely rotated around their fixation points on the nodes. If a node can be moved relatively to another node to another grid point, there is a degree of freedom. For an exact match, one or more degrees of freedom are undesired. Eliminating the degrees of freedom is done by introducing more edges. Ignoring the grid size, the green bolts in product $P_{4.5b}$ directly neighbouring bolt 20 can partially be rotated without generating new edges. The most right green bolts in product $P_{4.5c}$ are completely free to translate and rotate as they are not connected to the other bolts, but a local re-position of a bolt would result in a change in the graph. A degree of freedom is not present at bolt twelve in any of the products, since any rotation around this bolt would result in the generation of new edges or overlapping bolts. Note that the benefit of degrees of freedom is that imperfect matches would also be recognised, so potentially there is a sweet-spot of number of degrees of freedom.

It would be beneficial if the graphs would be planar, which means having no crossing edges, to keep the number of edges low. Consider a 2×2 bolt cluster as depicted in Figure 4.6, with the example in Figure 4.6b being rotated 90° anti-clockwise in relation to the example in Figure 4.6a. It is clear that crossing edges exist due to the layout of set of bolts, when all edges are generated. Could one of the crossing edges be left out to make the graph planar? First consider the example in Figure 4.6a. If only the green-yellow edge is generated to make the graph planar, there is no full match with the layout of the example in Figure 4.6b with the blue-red edge. Mathematically, this non-unique triangulation is explained by the non-uniqueness of the Delaunay triangulation of the nodes represented by the bolts [40, 41]. A Delaunay triangulation represents the triangulation of a set of nodes with none of these nodes falling within any circumcircle of a subset of three other nodes. In this case, the circumcircle of any subset of three nodes out of the, in this case, four nodes falls on the same circumcircle, which makes the triangulation non-unique. Non-uniqueness is unacceptable, as the matching of two equal bolt layouts

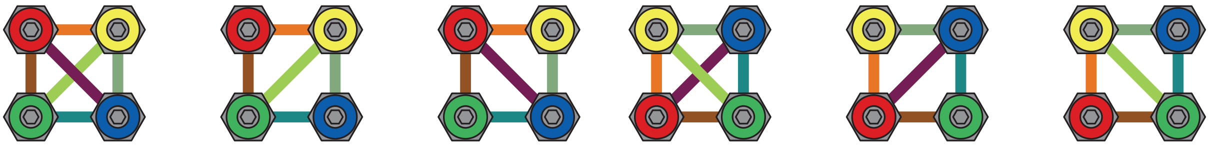


(a) graph representation of product $P_{4.5a}$

(b) graph representation of product $P_{4.5b}$

(c) graph representation of product $P_{4.5c}$

Figure 4.5: Graph representation of products



(a) A 2 × 2 bolt cluster example with all edge options depicted

(b) 90° counter-clockwise rotation version of example depicted in Figure 4.6a

Figure 4.6: Examples of non-unique graphs

may fail if two non-matching triangulations are generated. Two solutions are proposed. Both edges are generated, which means stepping away from Delaunay triangulation, or generating all possible graphs with non-crossing edges. If non-uniqueness is present more often, the number of graphs for one product increases exponentially, which makes comparing graphs of different products increasingly time intensive.

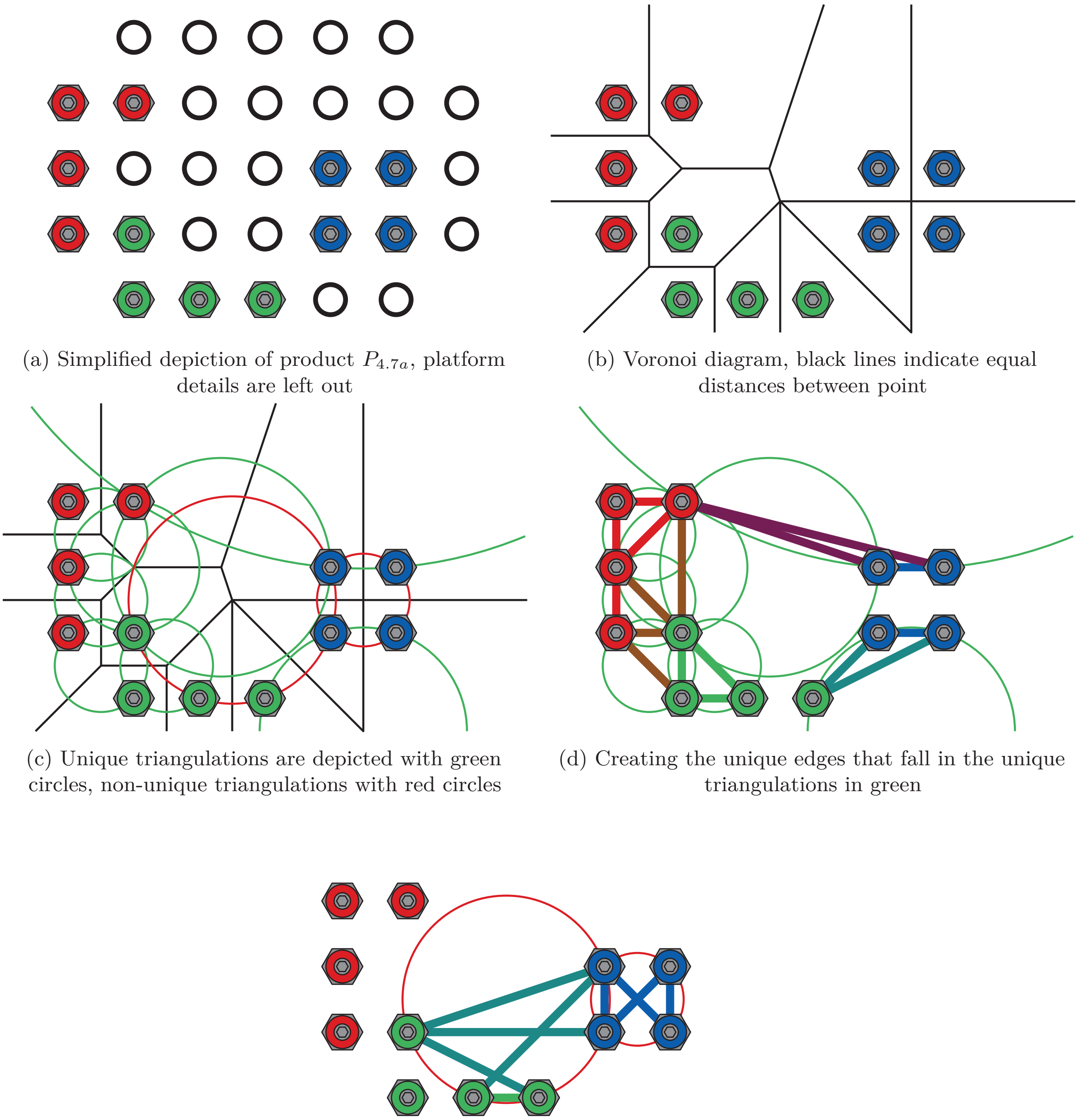
By defining which parts of the products can be represented by a Delaunay triangulation and which part cannot, a non-planar graph can be generated with a somewhat limited number of edges. This method is depicted in Figure 4.7. Figure 4.7a depicts the simplified product layout. In Figure 4.7b a Voronoi diagram is generated using the nodes represented by the bolts. A Voronoi diagram represents the lines on equal distances between nodes [40]. In Figure 4.7a the unique circumcircles are depicted in green, and the non-unique circumcircles in red. On the intersections of these lines, the midpoint of circumcircles is placed. The radius of the circumcircle is the distance between the midpoint of the circumcircle itself and the midpoint of the nearest node. If three nodes are located on the circumcircle, the graph is unique, if more than three nodes are located on the circumcircle, the graph is non-unique as not all nodes can be connected to each other without crossing. The part of the product that can be uniquely triangulated is depicted in Figure 4.7d and the non-unique part of the triangulation is depicted in Figure 4.7e.

A similarity measure defined by graphs can be created by comparing the graphs of two products in terms of their largest matching sub-graph. By creating different variants of the graphs, with different edge requirements, graphs can be generated that have various degrees of freedom that can be linked to non-unique products, to have a measure of similarity for less matching products. Varying the maximum edge length is one way of controlling degrees of freedom, as a shorter maximum length decreases the potential number of edges. The benefit of using graphs as opposed to clusters is the mathematical research that has already been performed into graph theory, as the problem of finding identical groupings of bolts present in one product in another product, is translatable to the sub-graph matching problem in graph theory [42]. Sub-graph matching, however, focuses mostly on the nodes. Therefore, an extension to incorporate edge lengths, or *weights*, should be investigated.

While graph theory has the potential to cover the mathematical side of finding common patterns in products, the question still remains how the operators actually see similarity in the products of the Omron setup. As the previously assembled products are limited in variation, no data is available to research the similarity definition for the products and operators beforehand. Therefore, similarity is forced into the products by introducing a set of pre-defined clusters of bolts, by which the products are buildup, called *blocks*. These blocks can be seen as the components of the products. Introducing a similarity measure for more variable products without the requirement of the limited set of pre-defined components is left for future research, with a graph based similarity measure recommended as research direction.

4.5 Block Based Similarity

All introduced similarity measures are defined without boundary conditions on the products with no guideline for the way similarity should and could be defined. While this allows much variability in the products, one could argue that for an initial exploration of the possibilities of self-learning in a high-mix and low-volume environment on the Omron setup, too much variability is undesired due to the negative effect on sample size of the experimental data. Therefore, components or building blocks out of which the products are built, are introduced which forces similarity to be present in the products at a higher level than the bolts itself.



(a) Simplified depiction of product $P_{4.7a}$, platform details are left out

(b) Voronoi diagram, black lines indicate equal distances between point

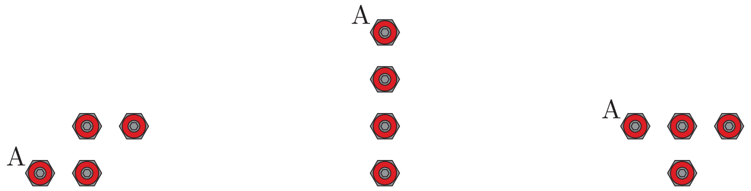
(c) Unique triangulations are depicted with green circles, non-unique triangulations with red circles

(d) Creating the unique edges that fall in the unique triangulations in green

(e) Creating the non-unique edges that fall in the non-unique triangulations in red, overlapping edges are left out

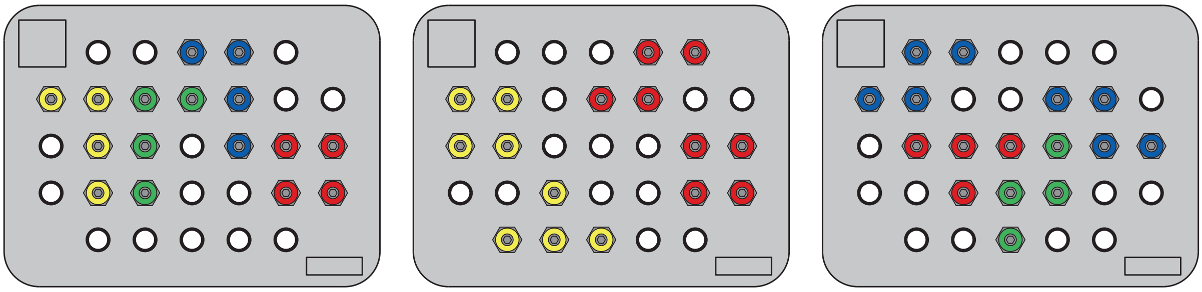
Figure 4.7: From nodes to edges

(a) O-shape (b) L-shape (c) L-shape (mirrored) (d) Z-shape



(e) Z-shape (mirrored) (f) I-shape (g) T-shape

Figure 4.8: Building blocks for the products, with index A indicating the origin of a block where position coordinates are based upon



(a) Product $P_{4.9a}$ (b) Product $P_{4.9b}$ (c) Product $P_{4.9c}$

Figure 4.9: Examples of products build from blocks

The building blocks are based upon the well known Tetris® block shapes; an O-shape (O), L-shape (L), mirrored L-shape (L(m)), Z-shape (Z), mirrored Z-shape (Z(m)), T-shape (T) and I-shape (I), each depicted in Figure 4.8. The bolt indicated with index A defines the origin of a block, which is used to describe the block's position. Each of the different blocks can be made with any of the four available colours. Mixed colours per block are not considered for simplicity. Blocks may not overlap, but may touch if the blocks are not of the same colour. Touching is defined as the distance between the two closest bolts of both blocks being equal to one. These conditions makes sure the blocks are clearly distinguishable, unique and recognisable. By introducing the blocks, similarity between products is present and is possible to be described by features of these pre-defined shapes. Variability is still present by defining enough pre-defined shapes, but capped by staying within reasonable amounts. To give a measure of the amount of variability possible, Table 4.2 gives an overview of the blocks and their feature options. The features are block shape, block colour, block orientation, block x-coordinate and block y-coordinate. With seven block shapes, four colours and four blocks per product, there are $(7 \cdot 4)^4 = 6 \cdot 10^5$ different options, not even taking into account the various positions and orientations the blocks can each have.

Table 4.2: Number of block features

shape	**O-shape**	**L-shape**				**Z-shape**		**I-shape**		**T-shape**			
orientation	0°	0°	90°	180°	270°	0°	90°	0°	90°	0°	90°	180°	270°
positions	20	16	17	16	17	18	16	10	16	19	16	19	16
mirror variant	no	yes				yes		no		no			
colours	red, green, blue, yellow												

It is expected that the operators recognise the shapes as being repeating patterns within assembly. The principles behind recognising shapes are introduced using the Gestalt principles in section 3.3.1. The common region and proximity gestalt principles seem very applicable to block based similarity, as the blocks are defined by common bolts, and separated by non-proximity of bolts of the same type. Two products consisting out of the same building blocks are at least similar in a way that a human is expected to recognise that a product is indeed constructed out of the same subset of building blocks.

Figure 4.9, depicts products $P_{4.9a}$, $P_{4.9b}$ and $P_{4.9c}$. These products can be translated into the form as presented in Table 4.3, which represents the products using the block shapes, colours, orientations and positions in x- and y-coordinates, also referred to as the block features. Orientations are limited to the meaningful ones, so only increments of 90° as any other value distorts the shape. Also, for example the O-shape, a 2×2 bolt shape, is equal in a orientation position of 0°, 90°, 180° and 270°, so no rotated variant is used. Similarly, for the I-shape and Z-shapes, only two rotations are used. This product notation makes it possible to compare products based on their features based upon blocks. Also, an analysis of the assembly performance based upon features can be performed to determine possible relations between the features and operator performance. All blocks consist out of four bolts, which makes direct comparisons possible.

Table 4.3: Products summarised using block shape, colour, orientation and position in x- and y-coordinates of the origin of each block. The position origin of a block is defined with index A

product	$\mathbf{P_{4.9a}}$				$\mathbf{P_{4.9b}}$				$\mathbf{P_{4.9c}}$			
shape	L,	L(m),	L,	O	O,	T,	Z(m),	O	Z(m),	T,	Z,	Z
colour	y,	g,	b,	r	y,	y,	r,	r	b,	r,	g,	b
orientation	180°,	180°,	180°,	0°	0°,	180°,	0°,	0°	0°,	0°,	90°,	0°
x-coordinate	2,	3,	5,	6	1,	4,	4,	6	1,	2,	4,	7
y-coordinate	2,	2,	3,	2	3,	1,	4,	2	4,	3,	1,	3

The block based products introduce repetition in the products, which can be used for a statistical analysis. This overcomes the need of defining an actual similarity measure, while still fulfilling the initial reason of searching for such a measure: increasing sample size. Comparing the complete products based on layout similarity to link to (optimal) assembly similarity is therefore currently not possible, but locally assembly patterns can be analysed on block level. On product level, the feature values of the four combined blocks in a product could be combined to define a feature representation based on the blocks for the complete product. This is however not explored yet, since this would still require extensive research into the weights and effect of every feature (combination) on similarity.

4.6 Conclusion

Of all similarity measures discussed, the block based approach is the simplest, but also most limiting one. However, it checks all necessary boxes; similarity is present in the products and repetition is present to have a big enough sample size, while keeping the high-mix and low-volume characteristics of the product. There is potential for an actual similarity measure, but this is currently not needed for the next step to take: comparing operator performance and assembly sequences.

A block feature based comparison is explored next to analyse whether there is statistical proof to be found that operators perform differently during the assembly of high-mix products. Based upon these results, the next steps are decided. In case these differences exist, there is potential to extract the optimal assembly patterns for operators. Chapter 5 introduces the experiment plan and discusses the observations and feedback gathered during the assembly sessions based on this experiment plan.

Chapter 5

Data Gathering

This chapter introduces the final product features and boundary conditions the product layout should comply to, to make sure the gathered data can be used for a performance comparison and analysis. Thereafter, the assumptions and expectations upon which the experiment plan is based are introduced. Knowing these assumptions, they can be validated in Chapter 6. Knowing the expectations, both behavioural and performance wise, it can be checked if they are true in actual assembly too. An initial experiment to validate operator behaviour is performed and analysed to make sure no fundamental assumption and expectation mistakes are made, which could make the results of the experiments beforehand less useful or difficult to analyse. Hereafter, the experiment plan is introduced. Finally, operator feedback concerning the final experiments is summarised, to know which statements made by operators should be validated in Chapter 6 and what possible improvements to the experiment and Omron demonstration setup can be made for future research.

5.1 Boundary Conditions

The data gathering, operators and data processing are bounded to the conditions listed below. The details concerning the experiment and assembled products itself are listed in section 5.5.

- Operators have to log in with a personal key card.
- Assembly is performed in the simulation mode of the Omron setup, which means:
  - The product recipe is only presented to the operator using the pick-to-light system.
  - Bolts are neither fastened or loosened by the Cobots.
  - The platforms stay in place.
  - Bolts have to be removed from the platform manually, and put back in the correct bins manually.
- The pick-to-light instructions only depict coloured circles where bolts have to be placed and depict the total number of every bolt colour to be picked on the respective bolt storage bins.
- The time between updating the pick-to-light system to the new assembly recipe and the operator actually starting assembly is not taken into account.
- Operators are explained that they are expected to assemble products without errors in a, for them, comfortable speed. They are expected to assemble the products to their best effort without mistakes, but are not rushed to do so.
- Operators are not told that the products consists out of multiple building blocks directly before the experiment, but some of them were aware of these blocks existing before starting assembly due to previous consultation about the research or involvement in the research.
- If operators are observed not assembling the products block by block, they are instructed to do so for the rest of the assemblies. After operator seven, this instruction was not given anymore. This decision is reasoned in section 5.5.1.
- Operators are not allowed to use the assembly platform as (intermediate) storage for bolts, bolts may only be in their corresponding storage bin, in the operators hand(s) or placed in the platform. This because of potential problems with the bolt recognition neural network and product quality problems in a real-life scenario.
- Operators are talked to during assembly by the experiment supervisor.
- Operators were instructed to not hang over the assembly platform while assembling to prevent blocking the depth camera and were instructed to not leave the safety curtain area while assembling due to observed problems with the pick-to-light system if done so.

- Operators are free to assemble the products to their own preference.

5.2 Assumptions

Some assumptions have been made concerning the products and assembly of the products, as summarised below.

- Operators recognise the building blocks from the pick-to-light instructions, as they are not outlined on the HMI or pick-to-light system, as depicted in Figure 2.5.
- New operators will need to get used to assembling on the machine which introduces a learning curve with natural assembly speed increase over the number of products produced.

5.3 Expectations

Based on discussions between Eindhoven University of Technology and Omron, there are certain behavioural expectations concerning the performance of the operators. The expectations are listed below.

- Assembly of a block is dependent on other blocks in the product.
- Few errors will be made due to the extensive assistance and easy assembly.
- Assembly order of blocks is dependent on their position in the grid.
- Assembly order of blocks is dependent on other blocks with the same colour.
- Operators will have directional preference of assembly, like working from left to right, top to bottom.
- Operators will have preferences of assembly per block, per colour or bolt by bolt.
- Operators will improve over time due to the expected learning curve.
- Operators will improve their assembly time of a product if they see it more often due to learning the product and improving their strategy.
- Natural decrease in assembly efficiency over time will be present after some time due to tiredness and disinterest of the operators.
- Operators will have preferences in assembly order within blocks.

5.4 Proof-of-Concept Experiment

As some uncertainties are present and no direct experiences are available of operators that assembled products on the setup who are not involved in the project, a proof-of-concept experiment was performed to check how a person who is not involved in the research performs. Here, block based products were assembled consisting out of four blocks. Here, it was clear that same coloured blocks may indeed not touch to still be recognisable, as here the no touching boundary condition of same coloured blocks was not implemented yet. Recognition of the repeating blocks in all products was not as clear as initially expected. The operator had no problems with assembling error free.

5.5 The Experiment

Every experiment consists out of a unique operator who assembles 42 block based products, with 16 of these being the same in every experiment, 16 unique for every experiment and ten of these being random repetitions of previous assembled products by the operator to have validation data of exact repetitions of assemblies by the same operator. If a repetition takes place, this always occurs twice directly following the first occurrence. So, an operator assembles a product trice for five times during an assembly session and all other products once. The non-unique products for the operators are all assembled in the first half of the assembly session, but always in different orders based on a balanced Latin square which distributes the number of times a certain occurrence follows or proceeds another occurrence, to minimise the carry over effect which would otherwise potentially be present [43].

Every product consists out of four blocks of each four bolts with no additional requirements on shape, location and colour than previously introduced except that the full set of random products is generated with an approximately equal amount of every block shape used by adapting the probability a certain block shape is chosen while generating the products. A purely random product generation method would not yield the desired normally divided results for a close to equal sample size for every product type, as not every block shape has an equal chance of actually being used in the product if chosen when the products are randomly generated. This is caused by a random location in the grid not having an equal chance of fitting every block shape, due to the layout of the blocks. The unique products are not checked for matches with the other product types, so a non-unique occurrence that is not a direct repetition would be random. Example products from the data set are already presented in Figure 4.9. A subset of the operators also assembled ten non-block-based, randomly generated products of equal size with four bolts of every colour for comparison to the structured products based on blocks.

The time intensity of gathering data, the current health circumstances in the world and therefore restricted number of people present at the location of setup, limited the possible operator sample size. However, based on results presented in Chapter 6, the current sample size of over 500 products assembled by twelve operators is viewed as sufficient.

5.5.1 Experiment Adaptions

When more and more operators assembled their product set on the setup, it became apparent that the recognition of blocks was not as straightforward as expected. Therefore, half way through it was decided to not require assembly block by block anymore, as operators still did it most of the time and at the time it was considered more interesting to investigate in which circumstances the operator would assemble block by block and in which circumstances they would not. During assembly, feedback was received concerning the difference of assembly performance and approaches when the bolts would not be clustered in blocks. As every product consists out of the same number of bolts, creating new products with the same number of bolts and colour occurrences in multiples of four would allow for a good comparison opportunity. Therefore, the final five operators assembled ten products with the random layout as previously introduced

5.6 Operator Feedback

Feedback of the operators is gathered during the experiments and after the experiments using a questionnaire. Below, a summary is presented with the main takeaways. Note that all statements in this summary are not facts or analysis results, but the statements of operators, which could be (unintentionally) false. The complete feedback list is provided in Appendix C.

Overall, it can be concluded that assembly is considered not challenging. Also, most operators did not seem to actively notice repetition, indicating a lack of actually processing what they were doing. Operators who assembled both the block based and random based products indicated that the clustering of the same coloured bolts in groups due the blocks, had a positive effect on assembly. Colours were often viewed as the deciding factor in the assembly order, with the bolts being in groups being of secondary importance. Bin order has been indicated as an order to follow, potentially being a consistent factor in otherwise random assemblies. Directional preferences were sometimes indicated. An abundance of a certain colour also seemed to influence assembly strategies. Preferences for or differences between any of the block shapes are not mentioned, with the exception of the Z-shape being indicated to be easier to assemble in parallel with two hands than the T-shape by one operator. Operators seemed to get the hang of assembly quite fast, so no significant improvement over time is expected. Random products seemed to be more of a challenge for operators, however, after some assembly realisations operators also seemed to get used to these.

During picking of bolts, operators differed between counting the number of bolts needed or just grabbing a hand full. If a hand of bolts was grabbed, most operators used one hand as intermediate buffer and placed the bolts with the other hand. The final two bolts could then be placed together. Returning bolts to the storage bins or not logically seems to correlate with respectively not counting and counting the bolts during picking.

Further overall remarks are that operators very rarely make an error. When they did, they where quite vocal about it as they did not understand why it happened. Quite often, bolts would fall during assembly due to movement errors during picking or placing. Bolts were retrieved after assembly of a product, not during. Concerning comfort, some of the operators indicated that the assembly platform was too high or too low, which was indicated as uncomfortable. Operators were also observed blocking the overhead camera sometimes, which is further discussed in section 6.1.3.

The most abundant remarks are translated into questions. These are summarised below and will be validated in Chapter 6.

- Is there a performance difference between block based products versus random products?
- Is there a a learning curve present?
- Does repetition have an effect on assembly strategies or performance?
- Which effect does an abundant bolt colour have on assembly performance?
- Does bin order have an effect on assembly performance?

5.7 Recommendations

The feedback concerning the easiness of assembly is interesting, but cannot be directly applied to the product as its nature cannot be changed easily. In reality, negative feedback concerning a product being easy to assemble would also not lead to a change of the product probably. However, it could be interesting to switch from the pick-to-light system to the HMI to allow the operators some alternation and more difficulty. Blocks consisting out of all four colours can be introduced to directly look into the effect of clustering of the colours and the difference of clustering of the bolts in the blocks versus random layout. Parallel assembly of platforms with custom pick-to-light instructions could potentially also increase assembly performance by grouping the bolts preferred by the operator together.

Concerning improvements to the setup, a relocation of the depth camera and beamer can be recommended. By putting it on an angle and further backwards, the operator will less likely block the sight on the assembly platform both allowing the neural network to more accurately track the assembly progression and the pick-to-light system to more consistently give feedback to the operator, as it would not be blocked as often. Potentially, the neural network can be extended by including a flag that switches value when too much of the platform is occluded, indicating potential incorrect bolt recognition. Furthermore, the bins can be adapted to allow dispensing the correct number of bolts and allow more structured alignment for easier picking. For a sense of progression, gamification, the introduction of a play element in assembly, can be considered by for example live comparison to other operators. However, operators should not be forced to overwork their abilities. Custom ordering of the bins could be interesting to investigate too, would operators like it or be confused by it?

Chapter 6 introduces how the assembly data is processed, after which the results are presented.

Chapter 6

Data Results

In this chapter, an overview is given of the results found in the data generated during the experiments discussed in Chapter 5. First, unexpected results not introduced in Chapter 5 are discussed, with if needed a description of how that result is dealt with in data processing. Hereafter, the importance of the differentiation between absolute versus scaled placement times is explained in relation to the learning potential of different operators, after which operator performance is summarised. Results with and without statistical differences between the analysed features are discussed to explore which features have potential for learning and which not. Also, some specific analyses are performed concerning starting preference, ending preference, the influence of dominant colours and performance differences between block based versus random products, as these topics were indicated to have influence on assembly performed in the feedback session. Hereafter, performance over time is analysed and clustering of the results is performed. These analyses give a clear overview of the potential of using the current available data of the chosen product type for a learning algorithm that can adapt instructions to improve performance. Thereafter, preliminary conclusions are made upon which a strategy based analysis is based in Chapter 7. Finally, preliminary recommendations are given for future research.

6.1 Unexpected Results

During the experiments some unexpected operator actions were noticed, which are summarised in this section. Problems with the data are also introduced, explained and a solution is presented if necessary.

6.1.1 Parallel Block Placement

As introduced, the features of the products are based on the components of the products, the blocks. It was expected that operators would recognise the blocks with ease and assemble the products block by block, which was not always the case. This makes comparing assembly times of blocks which are assembled in parallel not possible with the data available currently, as these times cannot be compared fairly to blocks which are assembled in serial. The reason for this is that only the bolt placement times are available, and not the times where the operator trajectory changes to the assembly of another block. Extracting motion primitives by tracking operator hand trajectories has the potential to introduce the data necessary to allow comparing these serial versus parallel assemblies as introduced in section 3.2.

The first seven operators were asked to alter their assembly strategy if it was noticed that they did not assemble block by block, as these assemblies would be unusable for data processing. However, based on feedback of previous operators, this was seen as quite limiting to the operator in hindsight. It was decided that enough data would still be available of block by block based assembly for the planned analysis, due to the previous operators and current operators also assembling block by block, but just not consistently. Finding out in what cases a certain strategy was used was considered more relevant than to keep limiting the strategy choices of the operators by forcing them to assemble block by block. These observations also led to the introduction and analysis of assembly strategies in Chapter 7 and Chapter 8.

6.1.2 Parallel Colour Placement

Similar to parallel block placement is parallel colour placement, however it can be seen as another action of operators. While parallel block placement can still be assembling a single colour bolt when multiple blocks have the same colour, assembling colours in parallel by definition means that blocks are assembled in parallel too. No further differentiation between these two actions of operators is made in this chapter. Chapter 7 does differentiate as in Chapter 7 strategies are compared.

6.1.3 Bolt Placement Tracking Problems

Due to the operator assembling in the space between the assembly platform and the camera, occluding the sight of the camera to the assembly platform happens easily. In this case, the bolts that are placed can not be tracked in real time, and will only be registered when the operator moves out of the way. Logically, this interferes with the assembly performance tracking. Assembly data of a block is not used if during its assembly more than two bolts are placed at the same time. While placing more bolts together at the same time is physically possible, it is not considered as normal assembly behaviour and is therefore viewed as a suitable threshold for marking an assembly of a block as incorrect. Note that not all boundary cases can be identified. For example, if a set of bolts is placed while the placement positions are occluded from the camera, these bolts are still detected sequentially if the obstruction moves slowly out of the way. Possibly, there is a bigger time jump between the previously last detected bolt placement, but this is no guarantee. As operators were asked to not occlude the camera, the only check implemented is the maximum number of in parallel detected bolts.

The neural network registers placements and removals of bolts, by logging the bolt index, bolt colour, time and action. Due to the occlusions, bolts are also registered as removed even when they in reality are still located at the correct placement position, as depicted in Figure 6.1. Bolts are depicted in their corresponding colours, with the removal operation depicted in a lighter tinted variant of the colours. It is assumed that bolts are not recognised while in a hand, and only recognised in a valid bolt position. Therefore, only the first occurrence of a bolt recognition in a certain position per assembly is used for further processing. A possible addition would therefore be checking bolts that are re-recognised: the neural network recognises previously detected bolts again if the fix to a bolt is lost, so if bolts are detected row per row from the back to the operator, this could also indicate an occurrence of an operator moving out of the way. This has not been implemented because not data is stored of the experiments which can confirm the correctness of this method.

6.1.4 Outliers

Outliers in the data are not specifically filtered, as during assembly no real occurrences of anomalies happened. The expected low number of occurrences of outliers is therefore deemed too insignificant for special algorithms and will be handled by the outlier identification of the main visualisation method in this research: boxplots. Boxplots are introduced in Figure 2.7.

6.1.5 Placement Errors

While assembly errors are rare, they do happen. During all assembly sessions, which corresponded to 546 assemblies, just six had errors. An error is logged when an incorrect bolt is recognised in an incorrect position. No bolts have been placed on positions were no bolt should be placed, all errors were caused by wrong colours or missing bolts. Note that the total of assemblies is not a multiple of 42 (or 52 if the random products were assembled too), as sometimes some extra assemblies were performed, for example for demonstration purposes.

Now that the problems are known, a start can be made with the discussion of the actual results of the experiments. First, the importance of the difference between absolute versus scaled placement times is introduced.

6.2 Absolute Versus Scaled Placement Times

Assembly performance is tracked by saving the placement times of bolts. These are offset to zero based on the first placed bolt of the assembly of a certain product. The placement time of the last bolt is consequently therefore equal to the assembly time of the complete product. Based on the placement

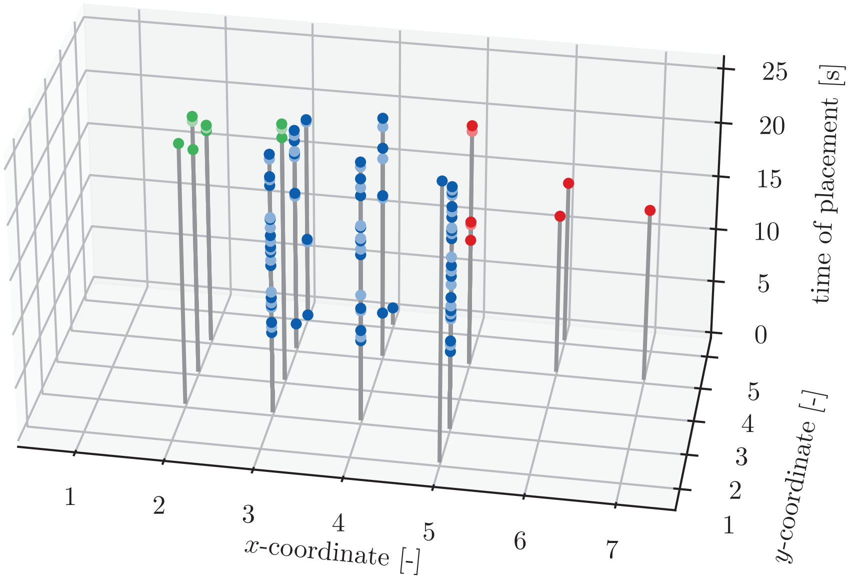


Figure 6.1: Raw assembly tracking data, with lighter tinted colours representing the removal of a bolt

times of bolts, the placement order of the bolts is known, which can be combined with bolt and or block features.

The problem with these absolute times is the performance difference between operators. If fictional operators A and B are compared with each other, and operator A is overall faster than operator B due to an arbitrary reason, their performance cannot really be compared to each other as operator A will always be faster than operator B in absolute time. However, operator A and B could still be similar in approach, or one could even be relatively better using an approach that the other does not use, which would make sharing assembly data between the two beneficial. Converting the times to scaled times makes the relative differences clear. Times are scaled by dividing through the mean value of the corresponding realisations. It is however an assumption that this extra analysis is necessary, and therefore both an analysis is performed with the absolute placement times and the scaled placement times. A variant of absolute placement time is relative placement time. Relative placement times refer to the time between the current placed bolt and the previously placed bolt. These clearer depict when bolts are placed together as the relative time would then equal zero. Bigger values could indicate the picking of bolts. Knowing the difference between absolute and scaled placement times and knowing why a distinction is made, the first results can be discussed next.

6.3 Operator Performance

Table 6.1 gives an overview of the detection success of the assemblies of all operators, to indicate how much data can be used from the total amount of data gathered. Note that these realisations refer to correctly recognised assemblies, in reality the operator could have assembled more. The detection success

Table 6.1: Overview of assembly detection success for all operators

operator		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
detection success	[%]	86.3	73.8	95.2	61.9	89.0	57.5	88.1	38.7	74.4	99.3	95.8	92.5

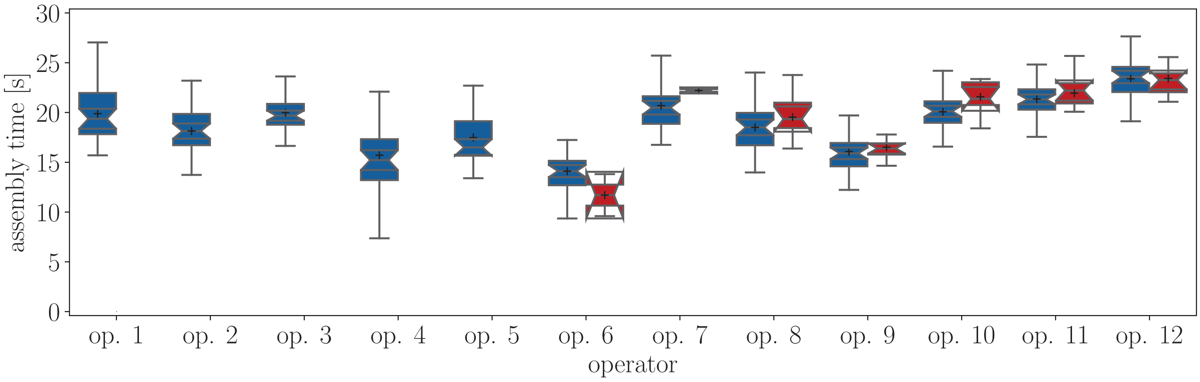


Figure 6.2: Assembly performance for every operator, with distinction between block based products in colour ■ and random products in colour ■

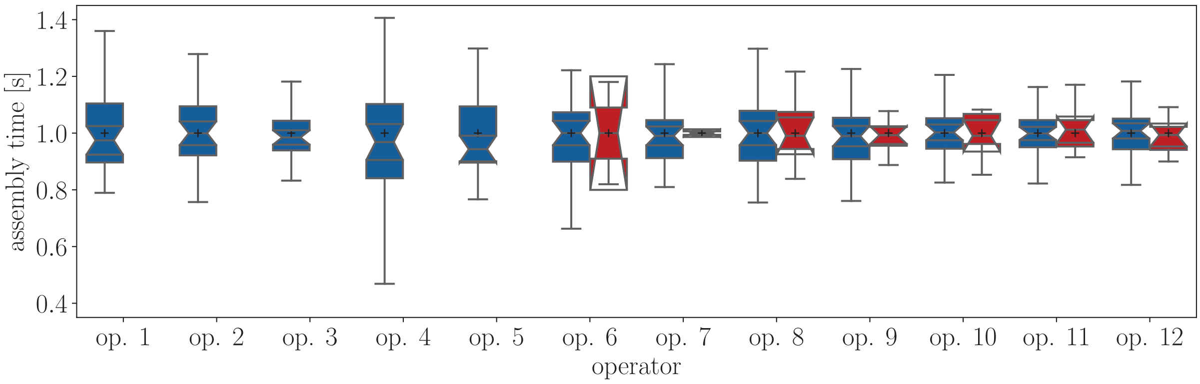


Figure 6.3: Scaled assembly performance for every operator, with distinction between block based products in colour ■ and random products in colour ■

is defined as

$$\text{detection success} = \frac{1}{4} \frac{\text{number of blocks detected}}{\text{number of detected products with at least one detected block}}. \tag{6.1}$$

The division by four is performed because every product consists out of four blocks. The denominator is defined as the number of detected products with at least one detected block because if no blocks are detected, no data of this product is available for analysis based on blocks. A detection success of less than 100% was to be expected by the feedback received concerning blocking the depth camera. Additionally, if operators do not assemble blocks serially, they are also not counted for the detection success as the block is not added to the database of block assembly data. Results in Chapter 7 do include products with their blocks placed in parallel. Table E.1 gives a detailed overview of how many realisations there are all block features.

The simplest comparison to be made is the assembly times for all products depicted against the different operators. Figure 6.2 depicts the absolute assembly times for all operators in boxplots, where clear differences between operators can be observed. The abbreviation *op.* stands for operator. A differentiation is made between the block based products and the random products, with block based products depicted in colour ■ and random products in colour ■. Overall, none of the operators are consistent in timing, but there are differences in the amount of spread. Overall, operators also have different assembly speeds, with the biggest difference between operators six and seven. In contradiction to the feedback, the assembly time of the random products seems slightly shorter than the block based ones. A deeper analysis is conducted in section 6.9 to take into account the learning curve of the operators, since all random products were assembled after the block based products. Note that operators six and seven only assembled two random products as a proof of concept, also given in Table E.1, explaining the respective extremely big and small confidence interval.

Figure 6.3 depicts the scaled assembly times where the overall differences between assembly speed are filtered out, leaving the spread of the assembly times. While it is clear differences are there, only operator four stands out with a relative high spread of assembly times. Overall, it can be concluded that operators have different assembly speeds and are not consistent in assembly speed. Therefore, an opportunity is present to find out why these inconsistent assembly speeds exist and find correlations between measurable features and the assembly performance. First however, the standardised colour coding for values of features are introduced, which are used in visualisation of the results.

6.4 Standardised Colouring for Operators and Block Shapes

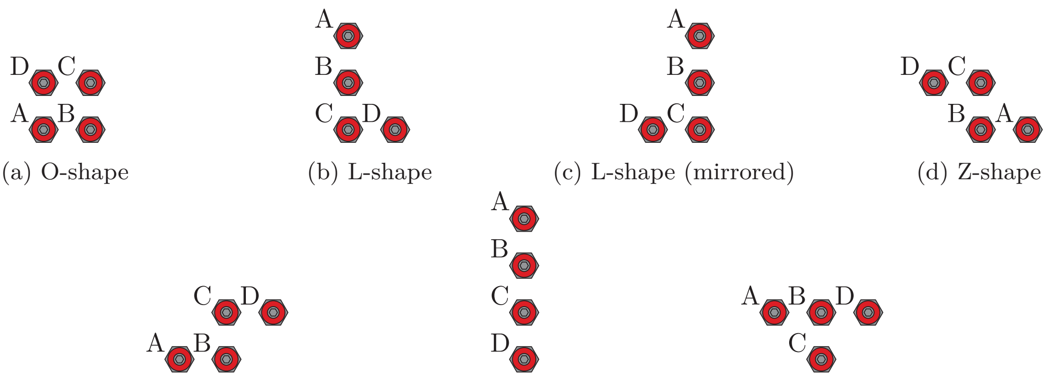
As the same features will be depicted repeatingly throughout the following sections, all visualisations where the colours differentiate between operators are sorted ascended left to right, with the operators indicated with the colours:

- operator 1;
- operator 2;
- operator 3;
- operator 4;
- operator 5;
- operator 6;
- operator 7;
- operator 8;
- operator 9;
- operator 10;
- operator 11;
- operator 12.

The block shapes are depicted similarly with consistent colouring, with O-shape, L-shape, L-shape mirrored, Z-shape, Z-shape mirrored, I-shape and T-shape, always depicted strictly in this order. Abbreviations are as defined in section 4.5. Note that the same colours can be used for different feature values as well, which is always indicated clearly. Within the blocks, the bolts can be uniquely identified, which is introduced next.

6.5 Unique Bolt Indices Within Blocks

All blocks consist out of four bolts, which allows $4! = 24$ different placement orders of bolts not considering parallel placement, which is a reasonable number of options to directly compare. Chapter 8 discusses this topic. The bolts within the blocks are given a unique identification letter, or index, to allow comparing the assemblies on block feature level. These are depicted in Figure 6.4. All blocks are depicted in their zero orientation position. Note that with exception of the L-shape versus the mirrored L-shape, and the Z-shape versus the mirrored Z-shape, the indices of the bolts are not to be compared between block shapes. If an O-shape and L-shape would both be assembled in the exact same order, this is a coincidence. The position of a block in x- and y-coordinates is defined for the bolt indicated with an A. Having defined the block indices, the placement times of bolts within a block can be compared based in placement order and on position within the block. The block indices also lay the foundation of local assembly strategies, based on exact placement orders within blocks, as discussed in Chapter 8. The first analysis results of the experiments are introduced next.



(a) O-shape (b) L-shape (c) L-shape (mirrored) (d) Z-shape

(e) Z-shape (mirrored) (f) I-shape (g) T-shape

Figure 6.4: Building blocks for the products with named indices

6.6 Performance Analysis Based on Block Features

To introduce similarity within the products, the bolt based component blocks are introduced in section 4.5. As a first step to find correlations between features of the blocks, the block features are depicted against the absolute assembly times using boxplots for the different operators. Every figure depicts one feature, with the colours indicating the corresponding operators as defined in section 6.4. First, the feature of the blocks' bolts being uniquely identifiable is introduced. Thereafter, the results without feature correlation to performance are presented to show that simply comparing the results of the features per operators is insufficient, after which more in depth analyses are performed.

6.6.1 Single Feature Analysis Per Operator

In this section, single features are analysed with operator performance depicted against operators, with the feature differentiating the boxplot visualisation using hues.

Performance Per Block Shape

Figure 6.5 depicts the performance of every operator versus the block shapes. Limited differences in operator performance within operators are expected as the blocks have different forms, but block orientation is also expected to influence the way operators assemble the blocks a lot, but is not yet taken into account here. Looking at the results there are clear differences between operators, related to the size of the spread of the assembly time, but the differences between block shapes for specific operators are not significant, with the exception of some, as the confidence intervals mostly overlap. Even when the confidence intervals do not overlap, as is the case for the comparison between the L-shape and the T-shape for operator five, the absolute difference is small. Differences between operators cannot be compared as it has already been shown that the mean assembly speeds of operators differ. Overall it can be concluded that if statistical differences in performance exist based on the block shapes, the other features also have an impact which causes a single feature analysis to not show concluding results.

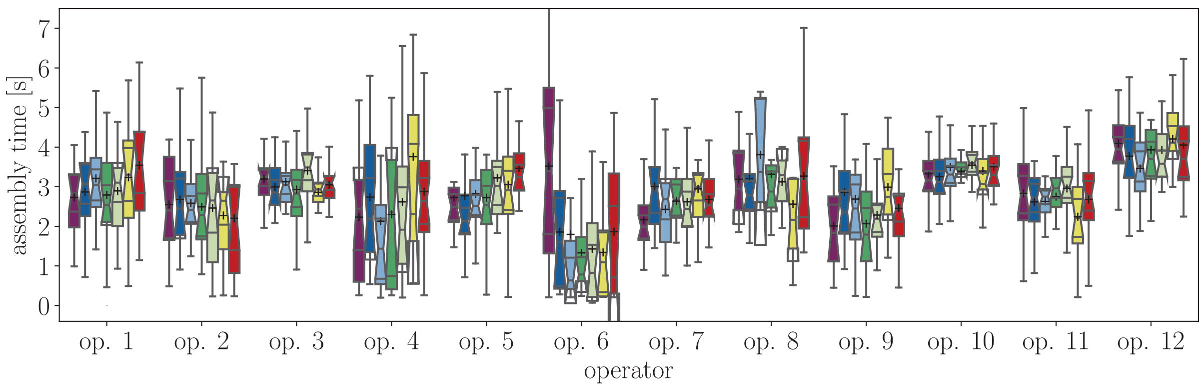


Figure 6.5: Assembly performance depicted for every operator, with hues from left to right representing the O-shape, L-shape, mirrored L-shape, Z-shape, mirrored Z-shape and T-shape. The O-shape block for operator six is capped for readability, actual value for the top whisker equals 9.68

Performance Per Block Colour

Figure 6.6 shows the comparison between the different block colours, with ■ red, ■ green, ■ blue and ■ yellow. No statistical differences in performance are expected as the colours should not influence assembly difficulty. Indeed, no statistical differences between colours are found for any operators. Potentially, only travel distances differ but these are filtered out if operators pick all necessary bolts beforehand, as the time starts when the first bolt of a block is placed.

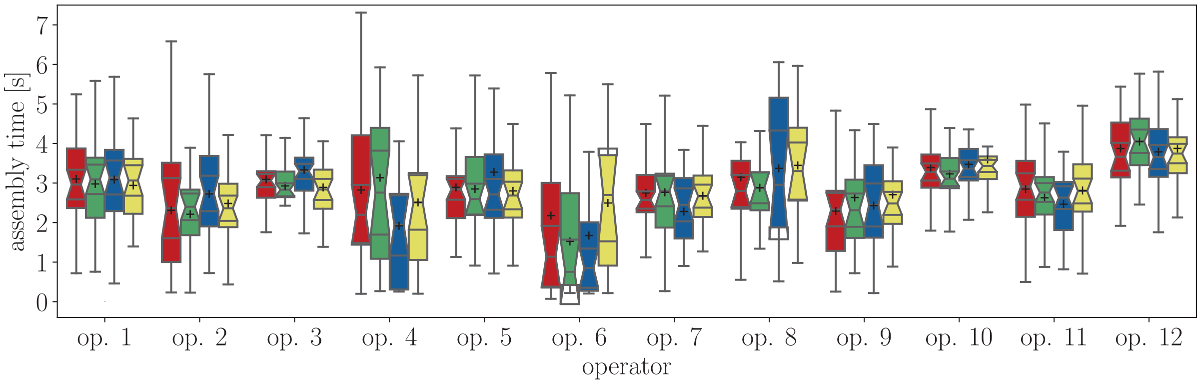


Figure 6.6: Assembly performance depicted for every operator, with hues from right to left representing red, green, blue and yellow

Performance Per Block Orientation

Figure 6.7 depicts the performance of every operator per block orientation. Colour ■ depicts 0°, colour ■ depicts 90°, colour ■ depicts 180° and colour ■ depicts 270°. Following a similar reasoning as for the block shapes being connected to the block orientation, block orientations are linked to the block shapes. The definition of the base orientation between two different blocks is defined arbitrarily, so a correlation between these would be by chance. Therefore, no statistical differences for a specific operator are expected. Indeed, no significant differences concerning the median are observed, as all confidence intervals within operators (almost) overlap. Again, the differences between operators are overall performance differences, so differences between certain orientations cannot be compared between operators.

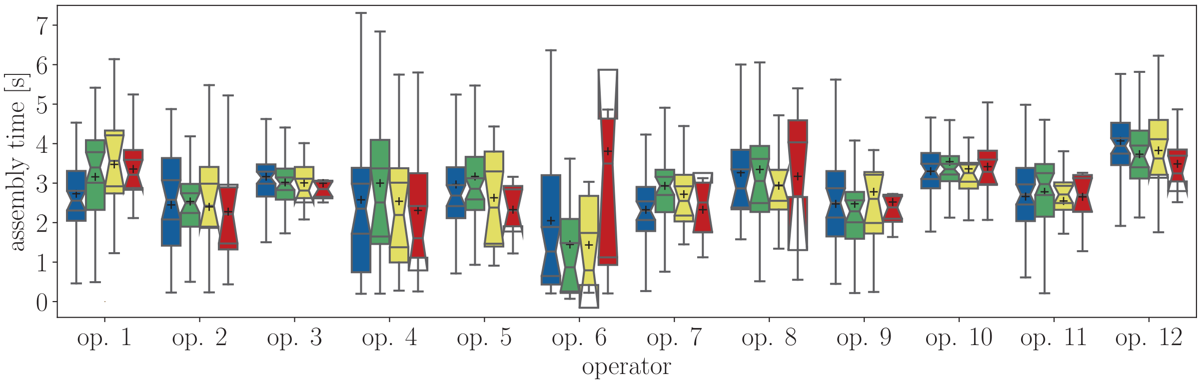
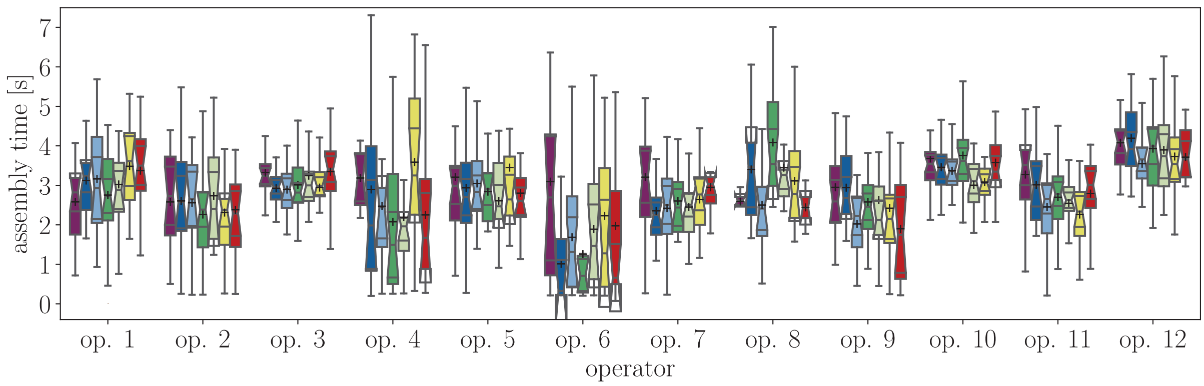


Figure 6.7: Assembly performance depicted against the different block orientations for every operator, with hues representing the orientations in increasing order from left to right

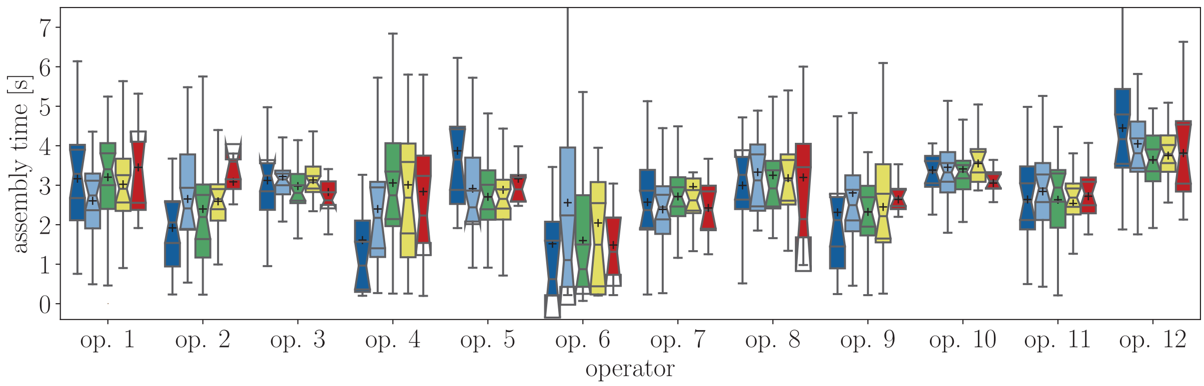
Performance Per Block Position

Figure 6.8a depicts the performance of every operator per block x-coordinate, based on bolt A. From left to right, colour ■ depicts $x = 1$, colour ■ depicts $x = 2$, colour ■ depicts $x = 3$, colour ■ depicts $x = 4$, colour ■ depicts $x = 5$, colour ■ depicts $x = 6$ and colour ■ depicts $x = 7$. As operators indicated that they work directionally, an effect of the block coordinates is expected in the order of assembly. However, as only the assembly times of the blocks are analysed here, no direct effect of the block coordinates in either x- or y-direction is expected. Potentially, the greater distance of lower y-coordinates to the storage bins does increase assembly time slightly. This effect is not expected to show in an analysis based on x-coordinates, as there is no differentiation between colours. Inspecting the data indeed yields that most of the differences are found in the spread, statistically the medians again do not differ much. Note again that inter operator differences are not to be compared, as the overall assembly speeds of the operators differ.

Figure 6.8b depicts the performance of every operator per block y-coordinate. Chronologically, colour ■ depicts $y = 1$, colour ■ depicts $y = 2$, colour ■ depicts $y = 3$, colour ■ depicts $y = 4$, and colour ■ depicts $y = 5$. The limited influence of lower y-coordinates increasing assembly times does not show statistically. Again, most of the differences are found in the spread, statistically the medians do not differ as the confidence intervals overlap.



(a) Assembly performance depicted against the different block x-coordinates for every operator, with hues representing the x-coordinates in increasing order from left to right



(b) Assembly performance depicted against the different block y-coordinates for every operator, with hues representing the y-coordinates in increasing order from left to right

Figure 6.8: Assembly performance depicted against the different block coordinates

6.6.2 Multiple Feature Analysis Per Operator

Until now, only the performance per block was of importance, but within the data of the blocks itself a lot of information is present too. Based on the feedback operators have given, it is known that operators assemble one bolt at a time, but also two bolts at once sometimes. Some operators pick only some bolts, while others pick lots. This should have an effect on the relative placement times between bolts, as it takes time to pick bolts. The grouping of bolt placements by operators is therefore expected. Operators would then place bolts in parallel, and jumps in time would occur when operators pick bolts. For this, an analysis is performed within the data of the blocks as presented in Chapter 8. Specific bolts within blocks are uniquely identified using bolt indices. In summary, with the data of every operator, comparisons can be made for every:

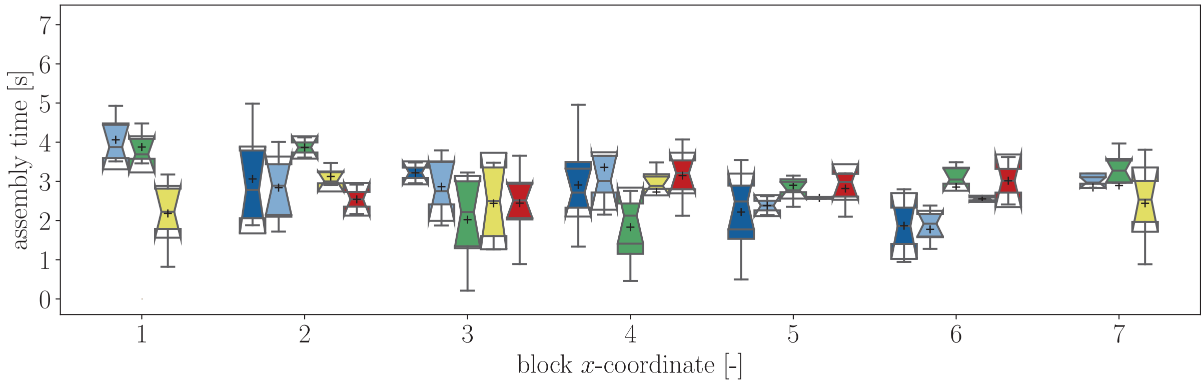
- block shapes, where hues represent:
  - block colours;
  - block orientations;
  - block x-coordinates;
  - block y-coordinates;
  - blocks' bolt indices;

- block colour, where hues represent:
  - block orientations;
  - block x-coordinates;
  - block y-coordinates;
  - blocks' bolt indices;
- block orientation, where hues represent:
  - block x-coordinates;
  - block y-coordinates;
  - blocks' bolt indices;
- block x-coordinate, where hues represent block y-coordinates or blocks' bolt indices.

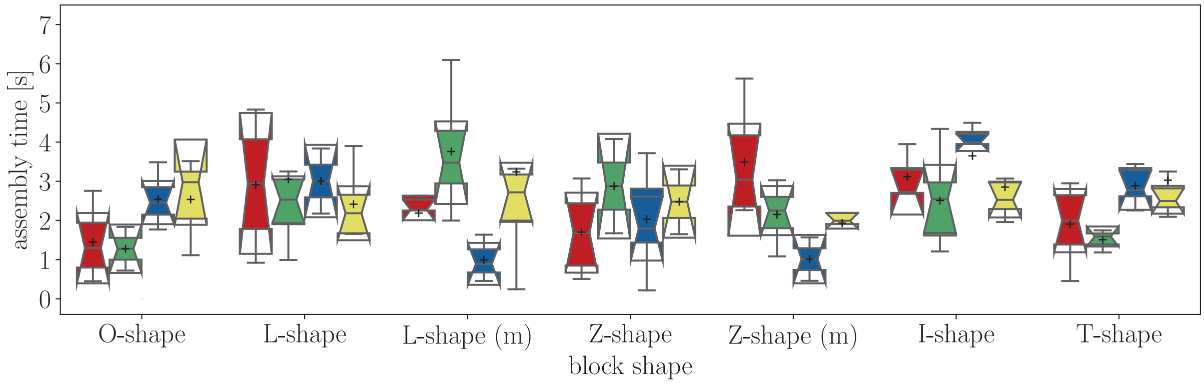
Note that swapping the feature on the x-axis and the feature represented by the hues, depicts the same data. Deeper analysis are made here where multiple feature values are combined and compared. Being so specific prevents assuming that a feature's values, which were previously not distinguished between, do not influence the response, but also reduces sample size significantly as more features values are locked. Figure 6.9a and Figure 6.9b highlight the results concerning block features with statistically the most distinctive results. Appendix E presents an overview of the combined value occurrences of features, from the in this chapter discussed operators.

Figure 6.9a depicts operator eleven's assembly performance versus the blocks' x-coordinates. The hues represent the blocks' y-coordinates. From left to right, colour ■ depicts $y = 1$, colour ■ depicts $y = 2$, colour ■ depicts $y = 3$, colour ■ depicts $y = 4$, and colour ■ depicts $y = 5$. The introduced effect of storage bin distance is potentially more apparent now, as now both the x-coordinate and y-coordinate is taken into account. The results show that the blocks in the middle position are statistically assembled with equal assembly speeds, while the outer blocks on low and high x-coordinates and low y-coordinates, so at the biggest distance of the storage bins, are assembled statistically slower. It should be noted that the sample sizes are low when features values are combined. Exact sample sizes can be referred to in Table E.8 for operator eleven.

Figure 6.9b depicts operator nine's assembly performance versus the blocks and the blocks' colours, which are represented by the respective hues, with respectively from left to right the colours red, green, blue and yellow. No statistical differences concerning assembly time based on this feature combination are expected as the block shapes and block colours are independent and both features on their own do not influence assembly time. Nevertheless, differences are observed. Again, these could be caused by the relatively low sample occurrences of the combined feature values. Exact sample sizes can be referred to in Table E.7 for operator nine.



(a) Operator eleven's assembly performance depicted against block x-coordinates, hues represent block y-coordinates in increasing order from left to right



(b) Operator nine's assembly performance depicted against block shapes, hues represent the blocks' colours red, green, blue and yellow from left to right

Figure 6.9: Assembly performance of blocks depicted against various features for various operators

Figure 6.10 depicts operator three's absolute bolt placement times of every block index. Index A is depicted in colour ■, index B in colour ■, index C in colour ■ and index D in colour ■. The block indices are independent of block orientation and position, as introduced in section 6.5. If operators have preferences to assembe a block in a specific way, this is expected to be recognisable by statistically unique orderings of the assembly order. However, due to the linkage of the block orientation and the block shapes, blocks with multiple orientation options are expected to need further specification of these feature values to see operator preferences. Based on the results, it is clear that operator three uses a relatively strict order within the blocks for the O-shape, as all confidence intervals are nicely separated with just a slight overlap of some whiskers. The same observation can be made for the I-shape, but as this block has a rotated variant, the limitation can be observed of this visualisation method based on a single block feature. It is clear that most of the times, operator three assembles in the order A, B, C, D, for both orientations the I-shape can have. However, if an operator would prefer a certain order in one orientation, and another order in the other orientation, the visualisation would look very different, while in reality there would be clear preferences. This hypothesis is applied in Figure 6.11a for an orientation of 0°, in Figure 6.11b for an orientation of 90°, in Figure 6.11c for an orientation of 180° and in Figure 6.11d for an orientation of 270°. Some combinations of block shapes and block orientations do not have sample occurrences, as not all combinations exist. Table 4.2 gives an overview of the possible combinations and exact sample sizes can be referred to in Table E.2 for operator three.

Figure 6.11a shows that the T-shape is also assembled in a strict order in an orientation of 0°, with the L-shape and Z-shape being less strict and with the mirrored L-shape and mirrored Z-shape not having a clear order of bolt placement. Note that the O-shape data is equal to the data shown in Figure 6.10, as this shape only has one orientation. Equal assembly orderings for different block shapes are a coincidence, as the block indices are allocated arbitrary between block shapes, with the only exceptions the allocation of the indices in the L-shape versus the mirrored L-shape and the Z-shape versus the mirrored Z-shape, as these are both mirrored versions of each other.

Figure 6.11b shows that in addition to the I-shape, the L-shape (m), Z-shape(m), and T-shape are also assembled in a strict order in an orientation of 90°. Orientations of 180° and 270° respectively depicted in Figure 6.11c and Figure 6.11d also show some strictness in placement orders, but note that the sample sizes have decreased significantly by setting multiple feature values as fixed, as can be referenced in Table E.1.

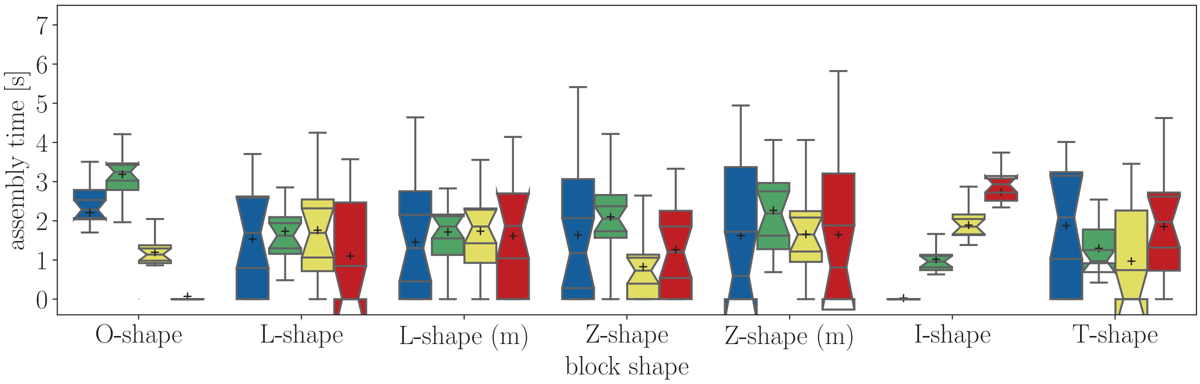
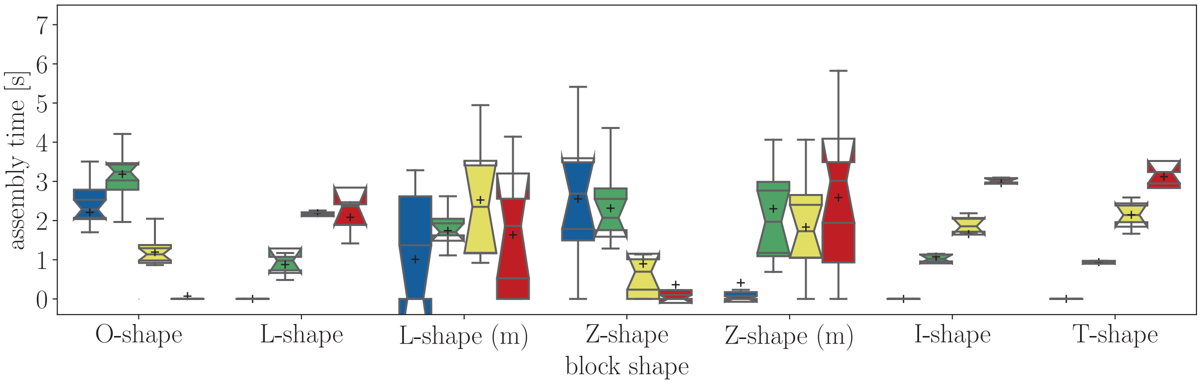
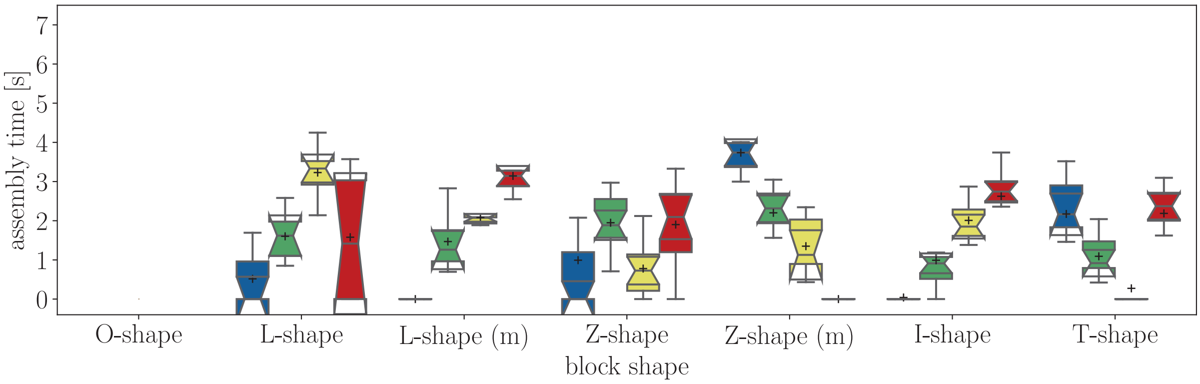


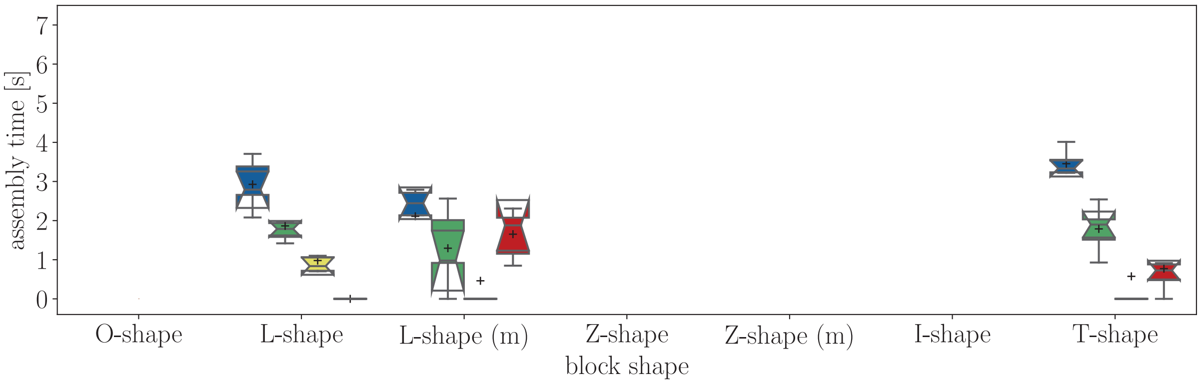
Figure 6.10: Operator three's assembly performance depicted against block shapes, hues represent block indices A, B, C and D from left to right



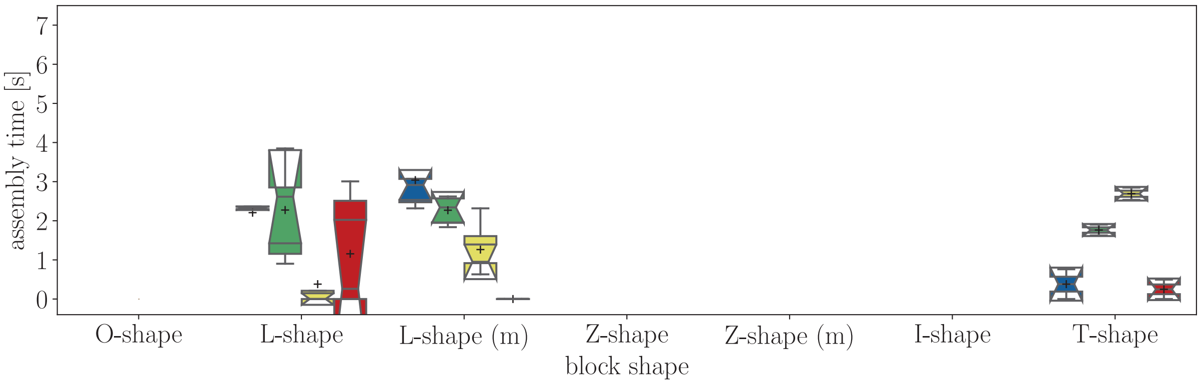
(a) Block orientation of 0°



(b) Block orientation of 90°



(c) Block orientation of 180°



(d) Block orientation of 270°

Figure 6.11: Operator three's assembly performance depicted against block shapes for various orientations, hues represent block indices A, B, C and D from left to right

Figure 6.12 depicts operator three's placement times of every bolt for every block shape, sorted in the placement ranking order. The first placed bolt is depicted in colour ■, the second placed bolt in colour ■, the third placed bolt in colour ■ and the last bolt in colour ■. This depiction does not sort the bolts' placement occurrences by block bolt index, but by placement rank and does therefore not represent the same data as the block index based visualisation. Now, the way bolts are placed should be easier to analyse, mostly to analyse serial versus parallel placement. Due to the uniquely identifiable placement order of the bolts as derived from Figure 6.11, operator three is expected to assemble serial, so bolt by bolt instead of placing two bolts at the same time. From Figure 6.12 it can be derived that operator three indeed works bolt per bolt, as the confidence intervals do not overlap. Note that the first bolt is always placed at an assembly time equal to 0. Figure 6.13 shows the same data but in relative time. In this representation, it can be more clearly seen what the time steps between bolt placement are. For example, a bigger difference between the second and third bolt could be expected if bolts were picked two at once, which is not the case here. Again, going deeper in the data based on for example the orientation of the blocks could yield different results based on the orientation of the blocks, but as the confidence intervals are clearly separated, this is not expected. The orientation options 0°, 90°, 180° and 270° are each respectively depicted in Figure 6.14a, Figure 6.14b, Figure 6.14c and Figure 6.14d. The general trend indeed seems equal and there is no indication of different assembly strategies. Only the spreads of the boxplots differ, probably caused by the differences in sample sizes.

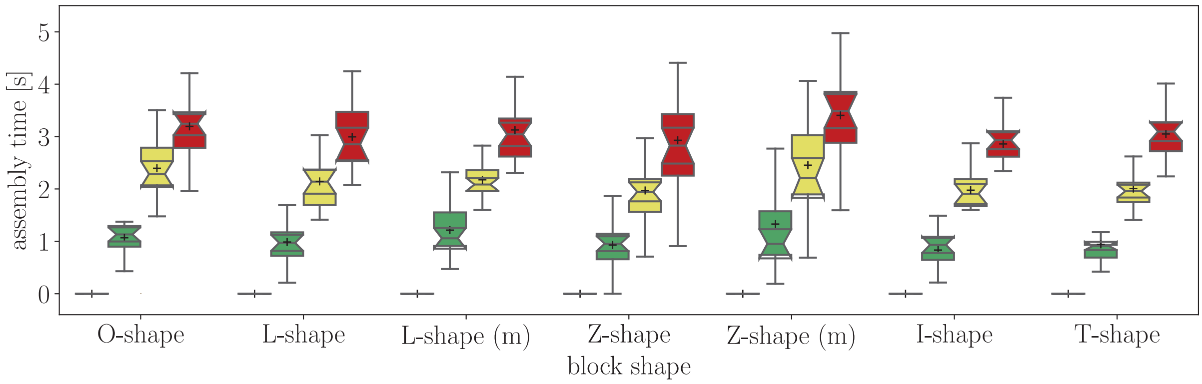


Figure 6.12: Operator three's assembly performance depicted against block shapes, hues represent a block's bolt placement ranking in increasing order from left to right

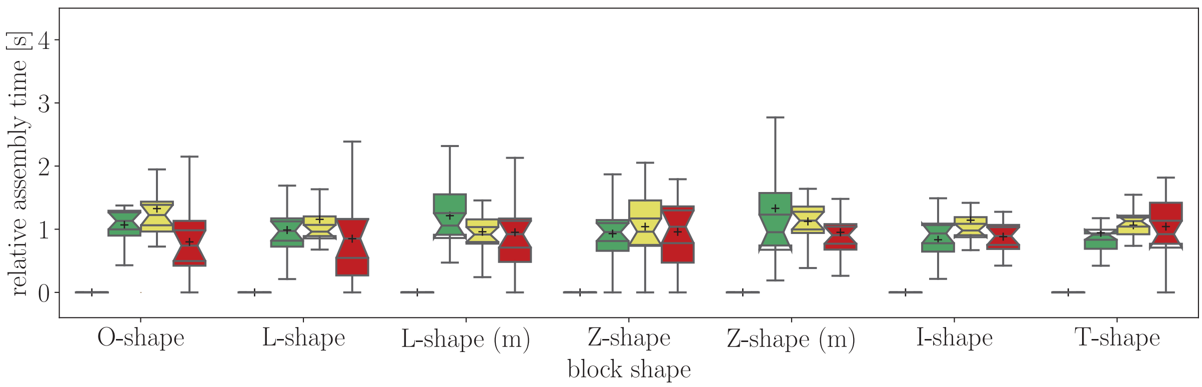
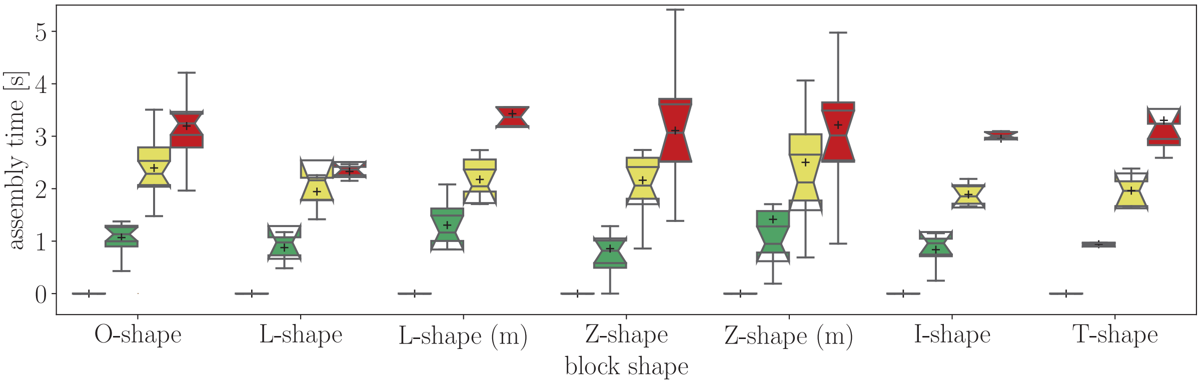
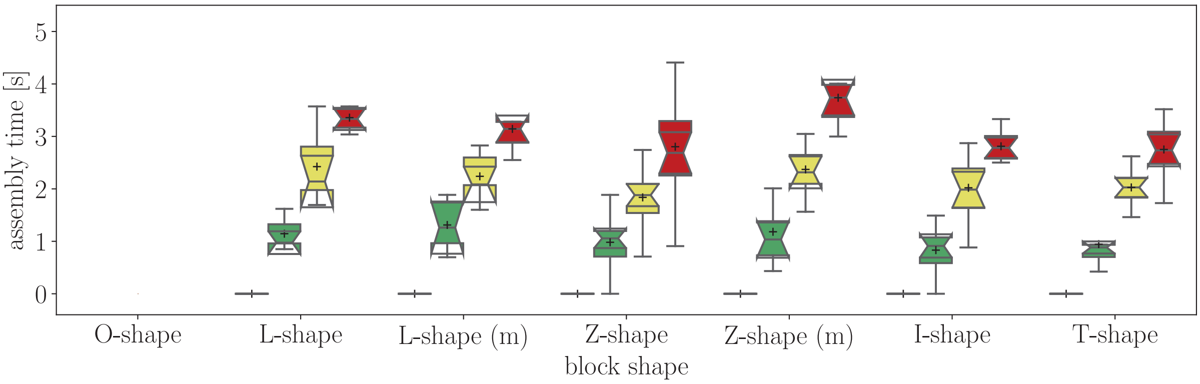


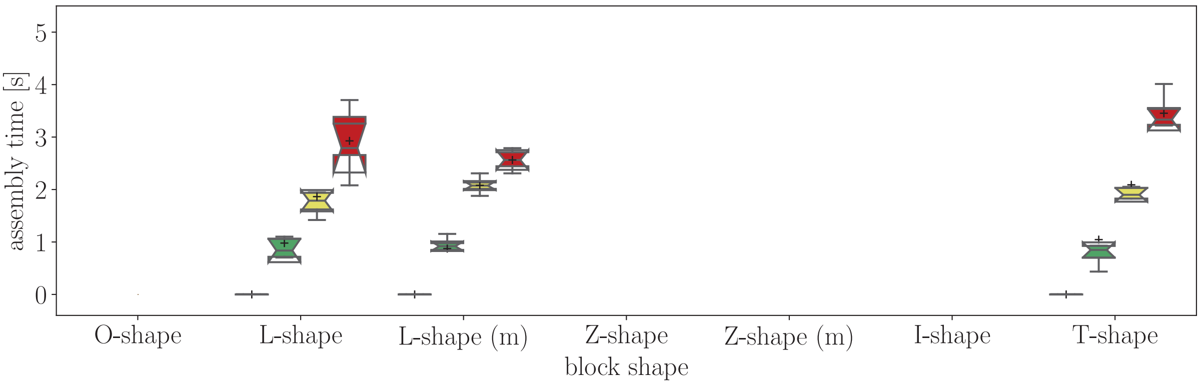
Figure 6.13: Operator three's assembly performance depicted against block shapes, hues represent a block's bolt placement ranking in increasing order from left to right in relative time



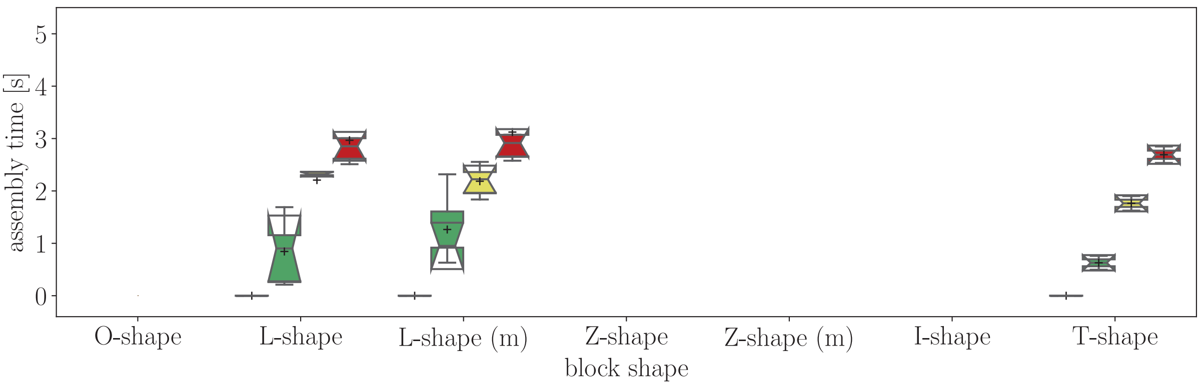
(a) Block orientation of 0°



(b) Block orientation of 90°



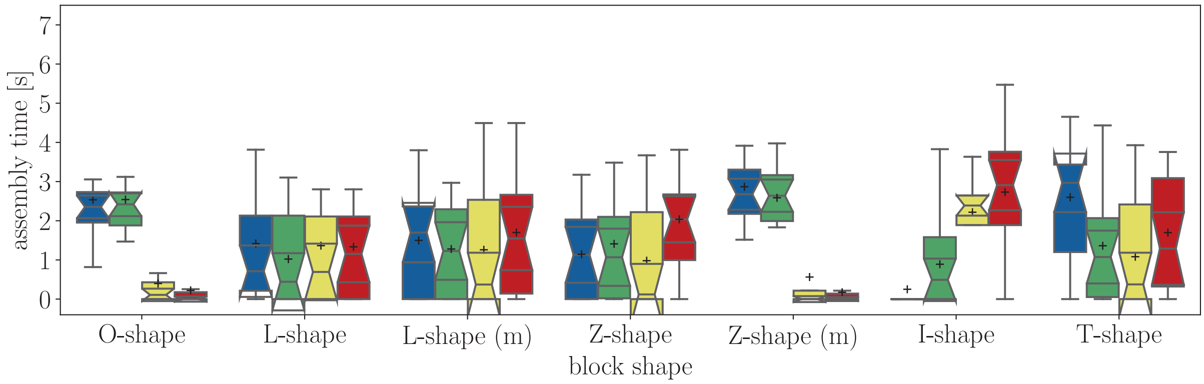
(c) Block orientation of 180°



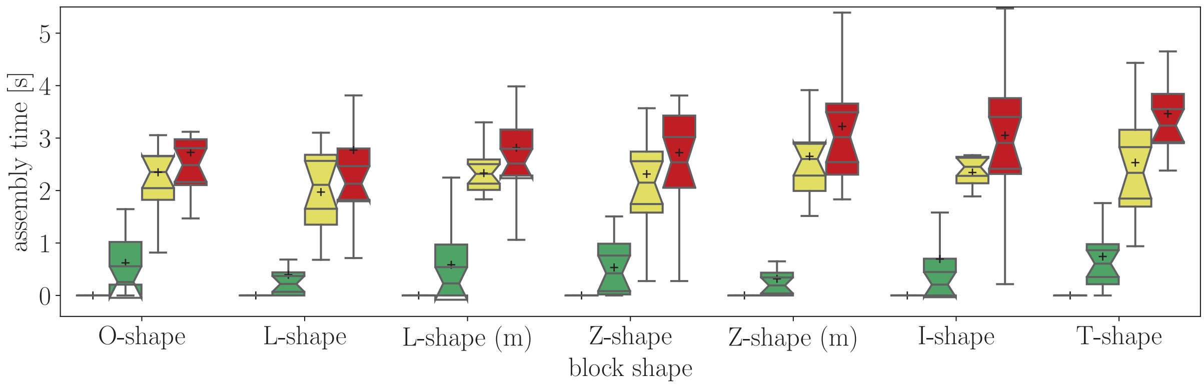
(d) Block orientation of 270°

Figure 6.14: Operator three's assembly performance depicted against block shapes with various orientations, hues represent a block's bolt placement ranking in increasing order from left to right

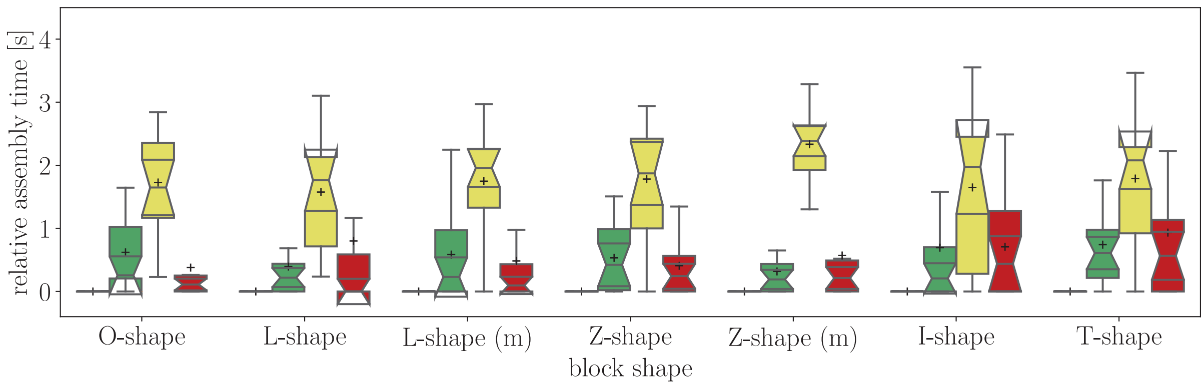
The same analysis as for operator three is performed for operator five. Figure 6.15a shows that there is quite a clear ordering for the I-shape, and for the O-shape and mirrored Z-shape it is clear that bolt C and D are placed first and second, and that bolt A and B are placed third and fourth. Figure 6.15b and Figure 6.15c confirm that the bolts are placed in sets of two. The other block shapes have no clear orderings, so again a deeper analysis additionally based on the orientations is performed. Exact sample sizes can be referred to in Table E.3 for operator five.



(a) Hues represent a block's bolt indices A, B, C and D from left to right



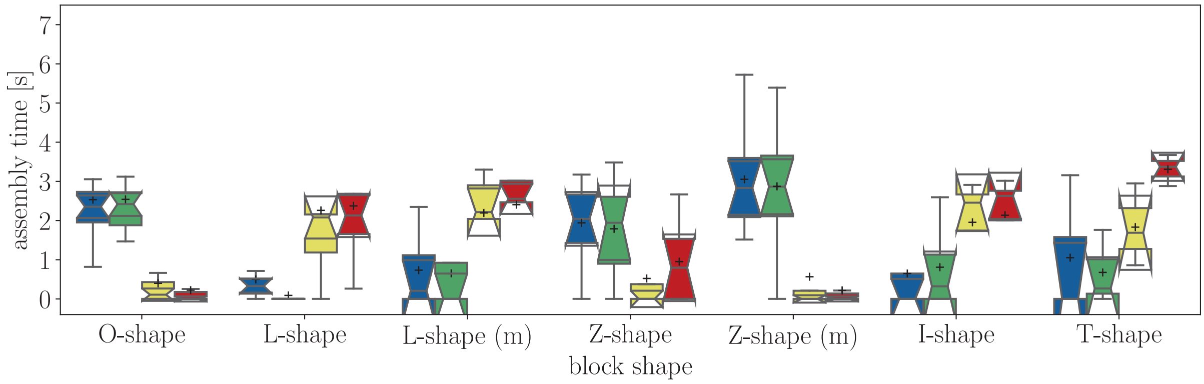
(b) Hues represent a block's bolt placement ranking in placement order from left to right



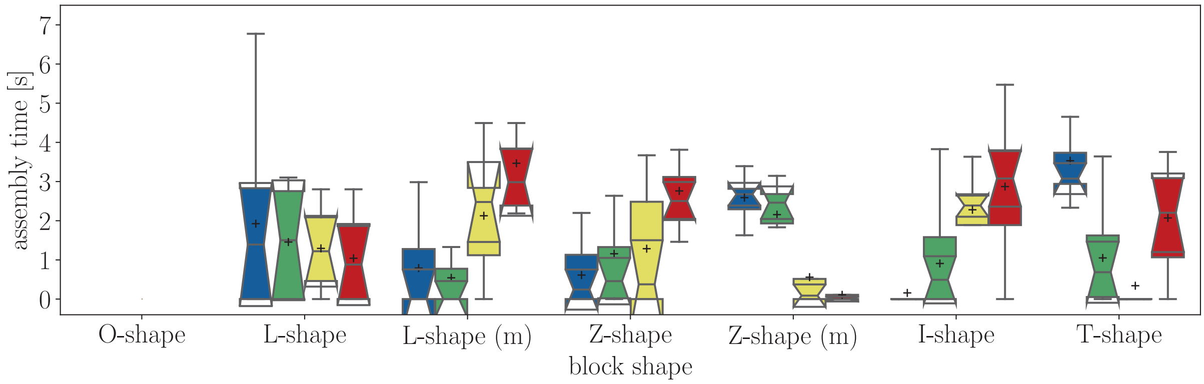
(c) Hues represent a block's bolt placement ranking in relative time in placement order from left to right

Figure 6.15: Operator five's assembly performance depicted against block shapes, with hues representing various representations of the placement times of the bolts within a block

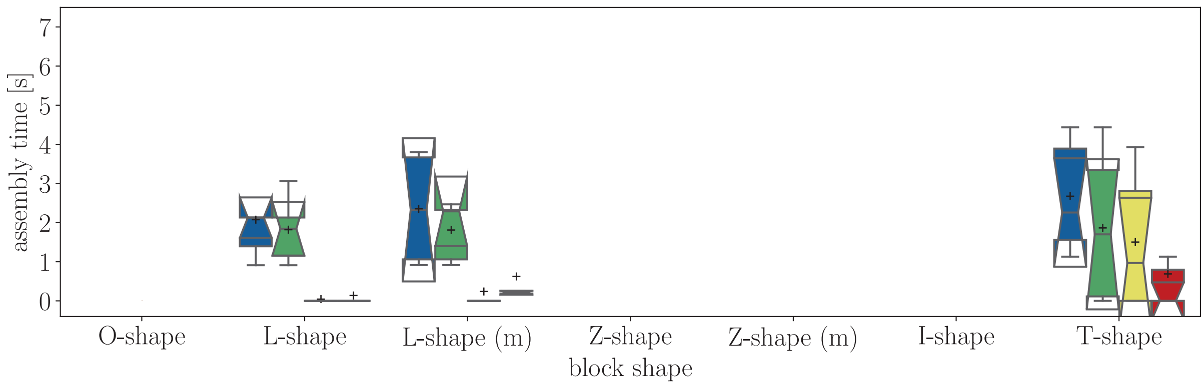
The orientation options 0°, 90°, 180° and 270° for operator five are each respectively depicted in Figure 6.16a, Figure 6.16b, Figure 6.16c and Figure 6.16d. The placement times of the blocks' indices A, B, C and D are depicted from left to right. For an orientation of 0°, it is clear which two bolts are placed together most of the time. With exception of the T-shape, each block shape has two sets of responses based on two indices have overlapping confidence intervals. For an orientation of 90° this distinction between the placement order of the indices is less clear for the L-shape, L-shape (m) and the Z-shape. Finally, for both the 180° and 270° orientations, the index order is less clear, expected to be caused by the limited data sample sizes.



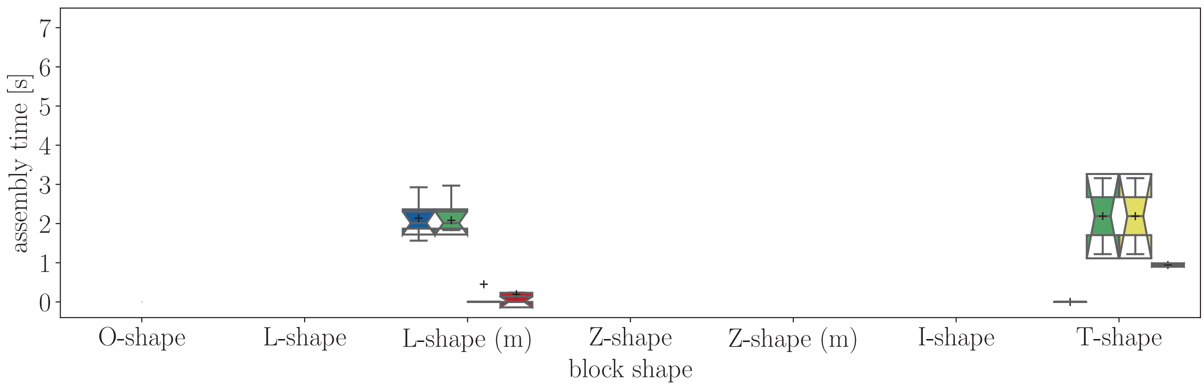
(a) Block orientation of 0°



(b) Block orientation of 90°



(c) Block orientation of 180°

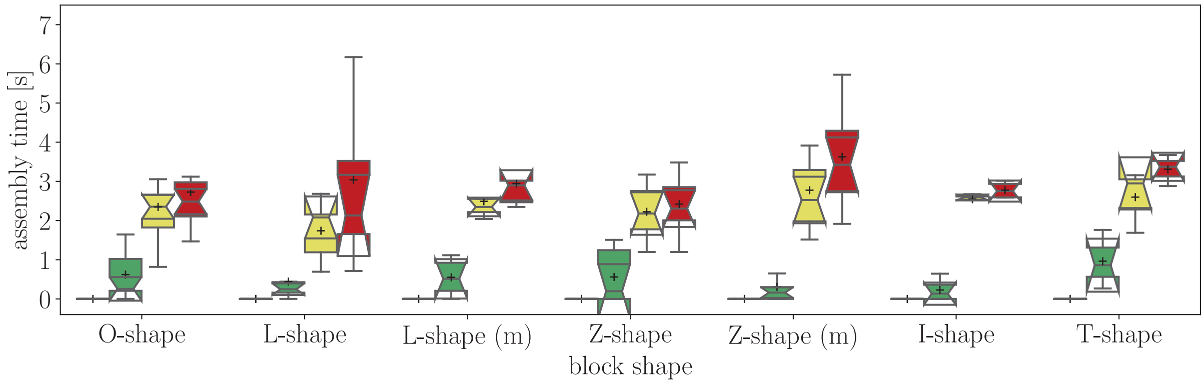


(d) Block orientation of 270°

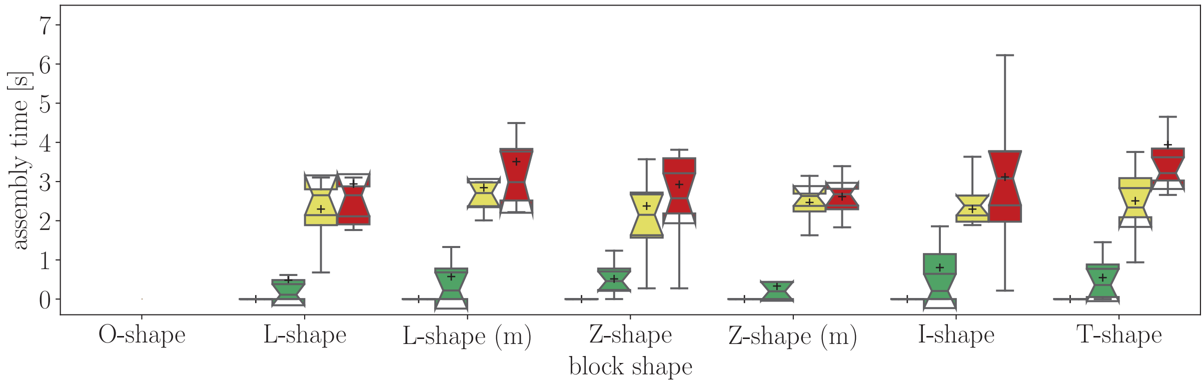
Figure 6.16: Operator five's assembly performance depicted against block shapes for various orientations, hues represent block indices A, B, C and D from left to right

A variant of the previous visualisations in Figure 6.16 is again based on the placement order of bolts. The orientation options 0°, 90°, 180° and 270° for operator five are each respectively depicted in Figure 6.17a, Figure 6.17b, Figure 6.17c and Figure 6.17d. The placement times of the ranking are depicted. It is clear already which bolts are placed together and what the overall ordering is, so no new observations are expected. Again the grouping of two is clear, with the T-shape again being an exception with a (nearly) non-overlapping confidence interval of the third and fourth bolt. Again note that the first bolt is placed at assembly time equals zero.

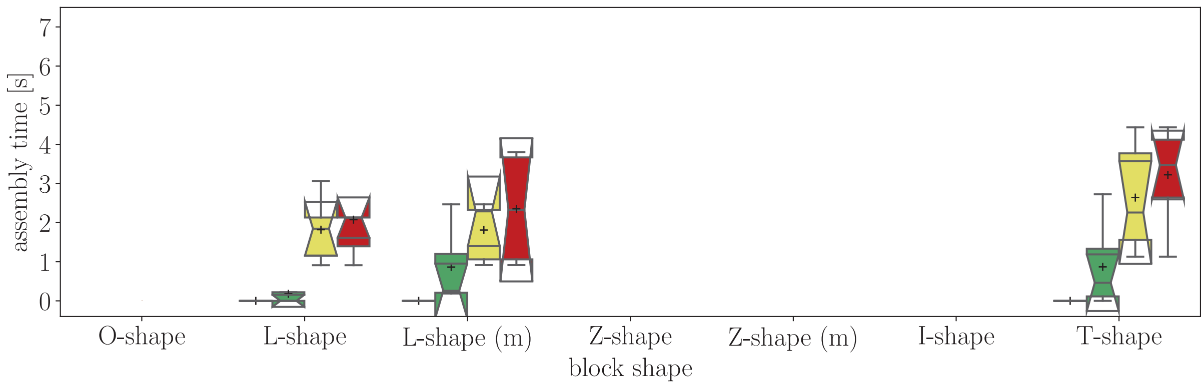
Overall, it is clear that operators have preferences for assembly order of blocks and have different timings to perform those orders based on either serial or parallel placement of bolts. A deeper analysis of the assembly orders of the blocks is performed in Chapter 8. The next step concerning overall performance analysis is analysing staring and ending preferences of operators of blocks, based on the blocks' features, as it was mentioned by operators that certain features had an influence on their assembly order.



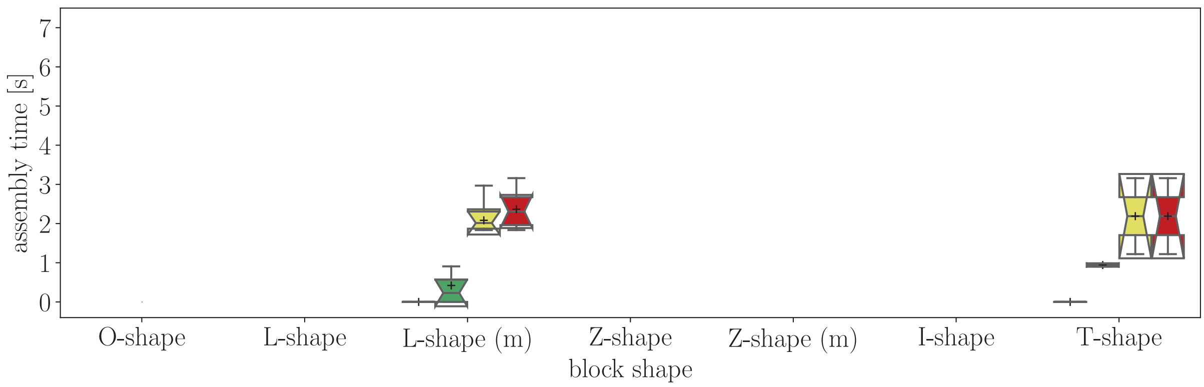
(a) Block orientation of 0°



(b) Block orientation of 90°



(c) Block orientation of 180°



(d) Block orientation of 270°

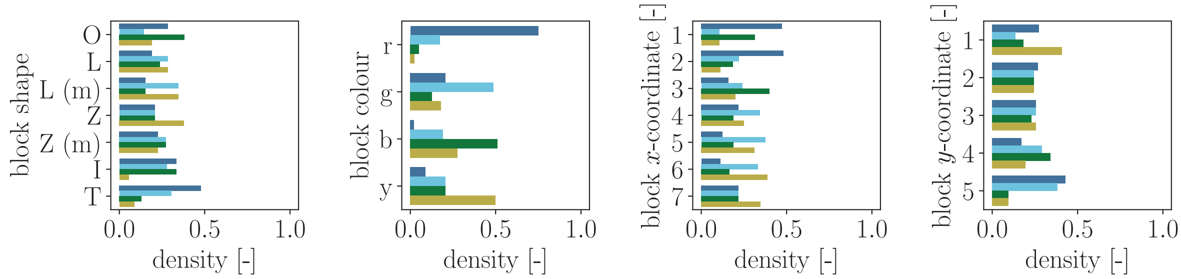
Figure 6.17: Operator five's assembly performance depicted against block shapes for various orientations, hues represent a block's bolt placement ranking in increasing order from left to right

6.7 Assembly Start and Ending Preference

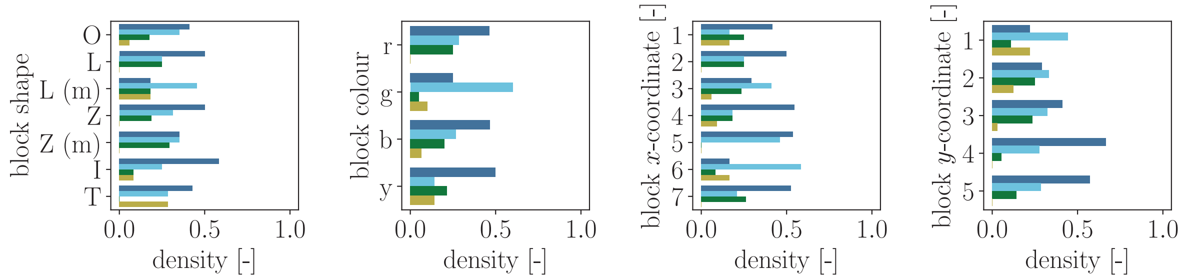
Feedback given by the operators raised the expectation that certain assembly preferences exist for starting and possibly ending of the assembly. Therefore, for every assembled product the number of times an operator assembled a certain block first, second, third and fourth based is analysed. This analyses is based on various features, namely block shape, block colour, block x-coordinate, and block y-coordinate. The distribution of occurrences is depicted using a density format, based on the number of times the block was placed on a specific order position based on a certain feature, divided by the total number of times a block based on that feature value was assembled by a specific operator. So the density value of a specific feature is the fraction of times a block with that feature value was placed in the indicated position order. Data of operator three, six and eight are respectively depicted in Figure 6.18a, Figure 6.18b and Figure 6.18c. The first placed block is indicated with colour ■, the second with colour ■, the third with colour ■ and the fourth with colour ■.

Assembly order is expected to be dependent on the colour red, as this is the most left colour in the storage bins, which was indicated by operators as a stable factor in assembly. In addition, directionality was indicated as having influence on the assembly order, as working in a defined direction also created some structure in assembly. For operator three, it can be seen that there is indeed a clear preference of starting with red, and a slight preference on starting left, as the density score of the first placed block on a low x-coordinate is high. There is a slight preference of starting topside of the product and also a slight preference of starting with a T-shape. Operator six has a clear preference of starting topside of the product. Operator eight has this same preference as operator six and additionally seems to start with the T-shape often if present. Starting with red is again preferred. Note that block orientation is left out as it is not a suitable indicator on its own, due to its dependency on the block shape. Leaving the block shape out is not considered, as operators should be able to distinguish block shapes independent on the block orientation, while the reverse is expected to not be the case.

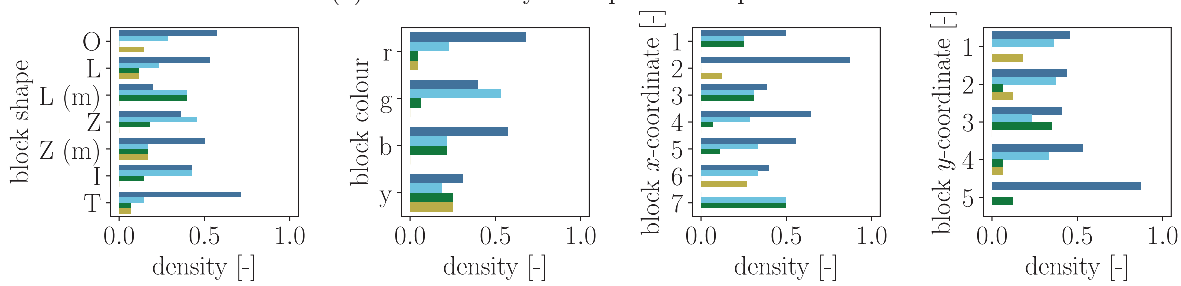
Continuing the focus on colours, colour dominance, which is defined as multiple blocks having the same colour, and its effect on operator performance is analysed next.



(a) Block assembly order preference operator three



(b) Block assembly order preference operator six



(c) Block assembly order preference operator eight

Figure 6.18: Block assembly order preference for selected operators. The first placed block is indicated with colour ■, the second with colour ■, the third with colour ■ and the fourth with colour ■

6.8 Comparison of Products with Dominant Colours

A specific case to be checked out is the influence of dominant colours within an product. The colour distribution within a product is random, so multiple blocks can have the same colour. An occurrence of that kind is described as a colour having dominance in received feedback. Figure 6.19 depicts the performance of products without dominant colours in grey, and the dominant products in their respective colours. Dominance is defined as a block with the same colour having more than one occurrence in a product. Occurrences are summarised in Table 6.2.

Table 6.2: Occurrence of colour dominance in products, with at least two blocks having the same colour

operator		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
dominance	[-]												
none		29	22	38	11	29	6	28	5	19	36	38	32
red		9	5	5	6	6	2	7	0	3	4	7	3
green		8	3	5	0	5	0	4	0	1	8	3	2
blue		5	6	12	1	9	1	6	0	5	9	10	9
yellow		5	4	6	2	5	3	3	2	4	11	7	9

While it is expected that colour dominance has an effect on the assembly order of products, this does not mean that it influences the assembly time. However, fewer different types of bolts are required to be picked which could decrease assembly time. Looking at the results, the spread of the no dominance occurrences is higher, but in similar scale as the original variance of all products as depicted in Figure 6.2. For operator two and nine there is some significant difference for dominant colours having a shorter assembly time, but overall their is no significant correlation of products with dominant colours being faster in assembly in comparison to products with no dominant colours. Further analysis of differences in assembly orders is therefore also not performed.

With colour dominance explored, the focus on colours is finished. Operator performance over time is explored next by fitting learning curve models on the performance data.

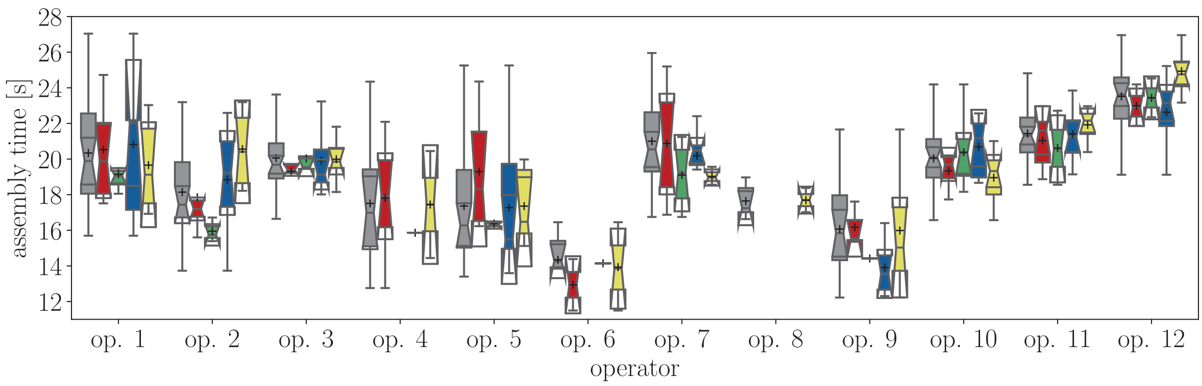


Figure 6.19: Comparison of performance of colour dominant and non colour dominant products, based on colours blue, green, yellow and red depicted from left to right

6.9 Operator Performance Over Time

Operators are all new to the assembly setup, so are expected to go through a learning curve while they progress through the products to be assembled. This learning curve can be recognised within the assembly time of the products, which is expected to decrease due to increased experience. When operators get more experience, they also may adapt their assembly strategy. This is however not analysed, as explained in section 7.5.

Learning curves are used to estimate the assembly time of operators taking into account learning [44]. It would be beneficial to know when operators reach a steady state so that assembled products during the learning phase can be disregarded in an eventual optimisation algorithm, as these products are not optimal. The learning curves can be represented with response T, which equals the assembly time per product, and $x \in \mathbb{R}^+$, the x^{th} product to be assembled. Various constants are used too: x_x, x_∞, λ, B, and C. A number of learning curve models are discussed by M.J. Anzanello et. al. [44], of which the suitable ones are selected to be fitted on the assembly data. As the assembly time of the operators is assumed to have a minimum value unequal to zero, the *Wright model*, $T = Cx^{-b}$ [44], cannot be applied due to its equal to zero limit. The *De Jong* [44] and *S-curve* [44] models are respectively equal to the *Wrights model* [44] and the *Standord B* model [44] in this use case. In addition to the remaining models, a linear and exponential model are fitted on the data too. The models are given below.

- Linear model
$$T = B + Cx \tag{6.2}$$

- Exponential model
$$T = (x_0 - x_\infty)e^{-\lambda x} + x_\infty \tag{6.3}$$

- Plateau model
$$T = B + Cx^{-b} \tag{6.4}$$

- Stanford B model

$$T = C(x + B)^{-b} \tag{6.5}$$

Fitting the models on the full assembly times of the products, gives the results depicted in Figure 6.20. Fitting is performed by least squares optimisation [9]. Only block based products are taken into account for fairness of comparison. The linear model is given by colour ■, the exponential model is given by colour ■, the Plateau model is given by colour ■, and the Stanford B model is given by model ■. Assembly time data samples are given by ● markers. Assembly times fluctuate and are visually not decreasing much, which explains the mostly quite rapidly decreasing response of the learning curve models. The data gets relatively constant quite fast and only has a few higher values at the start of assembly. Interestingly, operator nine seems to increase in assembly time over time, possibly caused by boredom.

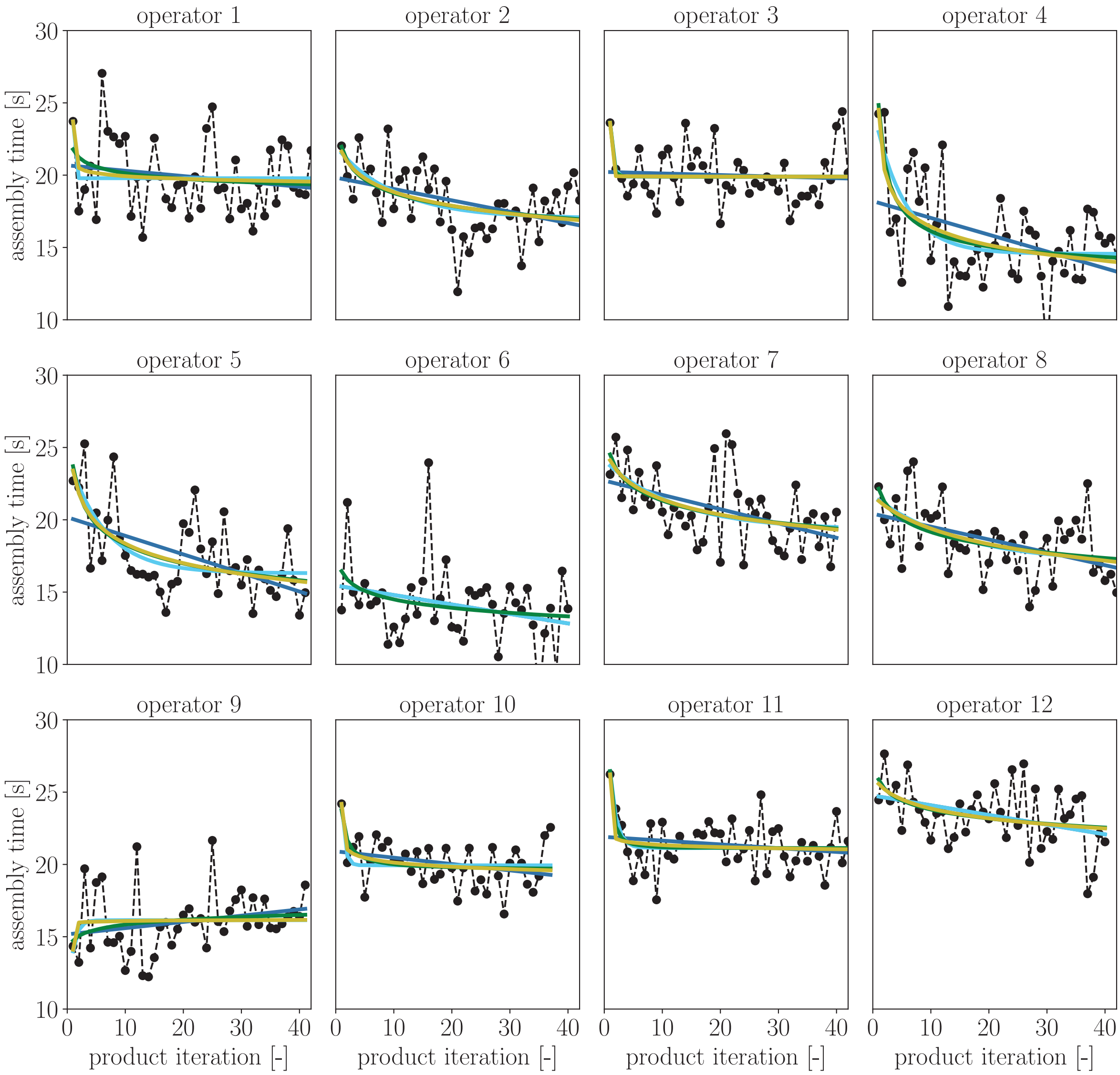


Figure 6.20: Learning curve models fitted over assembly time of products for all operators

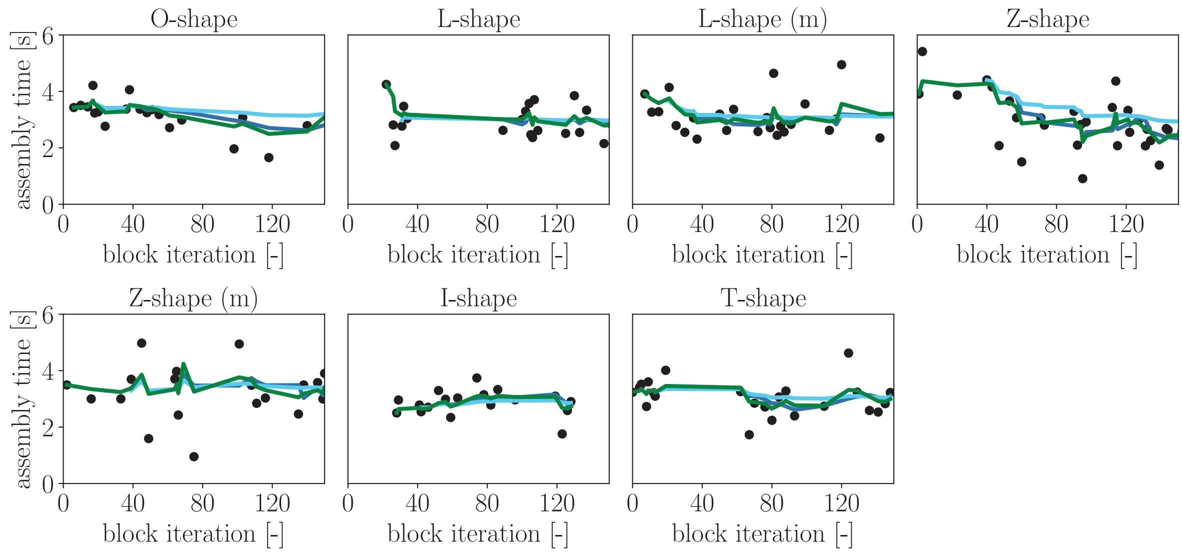
Figure 6.21a and Figure 6.21b analyse assembly time of blocks for respectively operator three and seven. For every block shape separately, the data samples are depicted with different types of averages fitted over it. Note that the big gaps in x-direction are therefore caused by that block shape not occurring for some time during assembly. In colour ■ the simple moving average with a window of seven is depicted, in colour ■ the cumulative moving average with minimum period of four is depicted and in colour ■ the exponential weighted average is depicted, with the weight factor of new observations equal to 0.3. As the full products doe not seem to have much decrease in assembly, and due to the expected added variance in the data due to the shorter assembly time of a block versus a product, no significant improvement over time is expected. Looking at the results, the overall trend of the samples does indeed seem to be that assembly gets faster, but only slightly. The variance in assembly times is however high in comparison with the actual assembly time, making it hard to observe trends if even present. Due to the limited results of the learning curves for the full products, the models are not fitted on the block data. Based on these observations, no data is disregarded based on its corresponding product or block order in the product set assembled.

The final step in analysing the results is clustering of the assembly performance response, and comparing the resulting clusters to the block features to see if there are correlations of feature values to the assembly time.

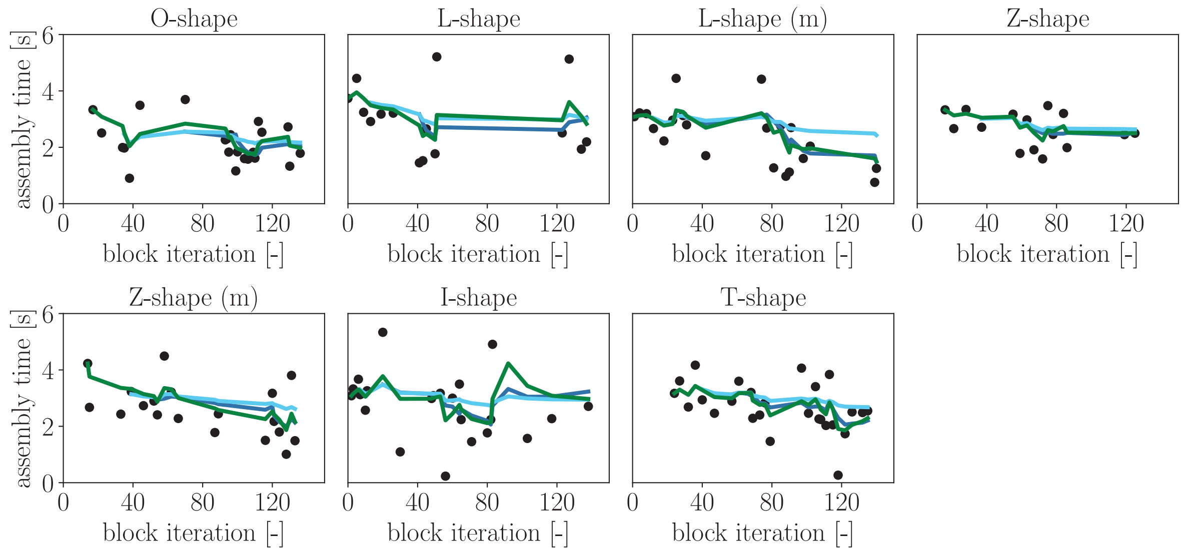
6.10 Clustering of Results

The assembly response data can be clustered, which potentially would create clusters with groups of block feature values. When the assembly response is used, the actual placement times, the differences between product layouts can be worked around and specific feature combinations can be found that together lead to fast or slower assembly performance. Therefore, clustering of the assembly data has the potential to find connections that are not pre-programmed. As an example, two operator characteristics are defined: the operator assembles bolts in red very fast, and assembles bolts in blue very slow. If the response is clustered and these clusters are depicted against the features, the slow assembles are expected to be mostly located at the block colour feature value blue, and the fast ones at block colour feature red. Additionally, clustering could also potentially define an extra uniformity of cluster allocation occurrences at a certain block type, if the block type also matters significantly for assembly time. The idea would be to eventually divide the assemblies in a, by the clustering itself determined number of clusters, based on assembly order and expressed in different features. These assembly orders are referred to as assembly strategies and are further explored in Chapter 7. Ideally, these strategies would also be depicted against the found clusters, to see if certain (combinations of) strategies are clustered together based on the assembly performance.

First a general overview is given. For operator three, consider Figure 6.22 for the absolute placement times of the blocks' bolt indices with generated clusters indicated by colours. Ideally the visualisation would have been depicted in four dimensions, but this is not possible so therefore a two dimensional representation is depicted by showing the six two-dimensional projections belonging to the four dimensional structure. Figure 6.23 depicts the placement times of the blocks' bolts in chronological order and Figure 6.24 depicts the placement times of the blocks' bolts in chronological order in relative time. As the bolt placements in chronological order and in relative time always have a time equal of zero for the first placed bolt, the placement times of bolt one are not depicted, and therefore the data is reduced to three dimensions. The distribution of the placement times are however added now for the other bolts. The clusters are generated using K-means clustering with a cluster amount of seven, which means that seven clusters are generated with midpoints which would not change if the closest data points to these midpoints are used to generate new midpoints. For more information concerning clustering, refer to the work of G. James et. al. [9]. The reason a cluster amount of seven is chosen, is because the maximum number of feature values equals seven. Therefore, all features theoretically can have their own cluster or a combination of multiple. More clusters is considered too many for manual analysis of the resulting clusters. The distribution of the absolute placement times of the bolts seems to have a clear most common value, with distributions similar to a normal distribution. The same analyses are depicted for operator five in Figure 6.25, Figure 6.26 and Figure 6.27 respectively. Operator five has very different distributions, caused by his parallel placement. Fitting know distribution models to this data could be beneficial for operator assembly time estimation like in the work of Cockx [7].



(a) Data of operator three



(b) Data of operator seven

Figure 6.21: Computed simple moving average (in colour ■), cumulative moving average (in colour ■) and exponential weighted average (■) of assembly times of all block shapes depicted against the number of assembled blocks

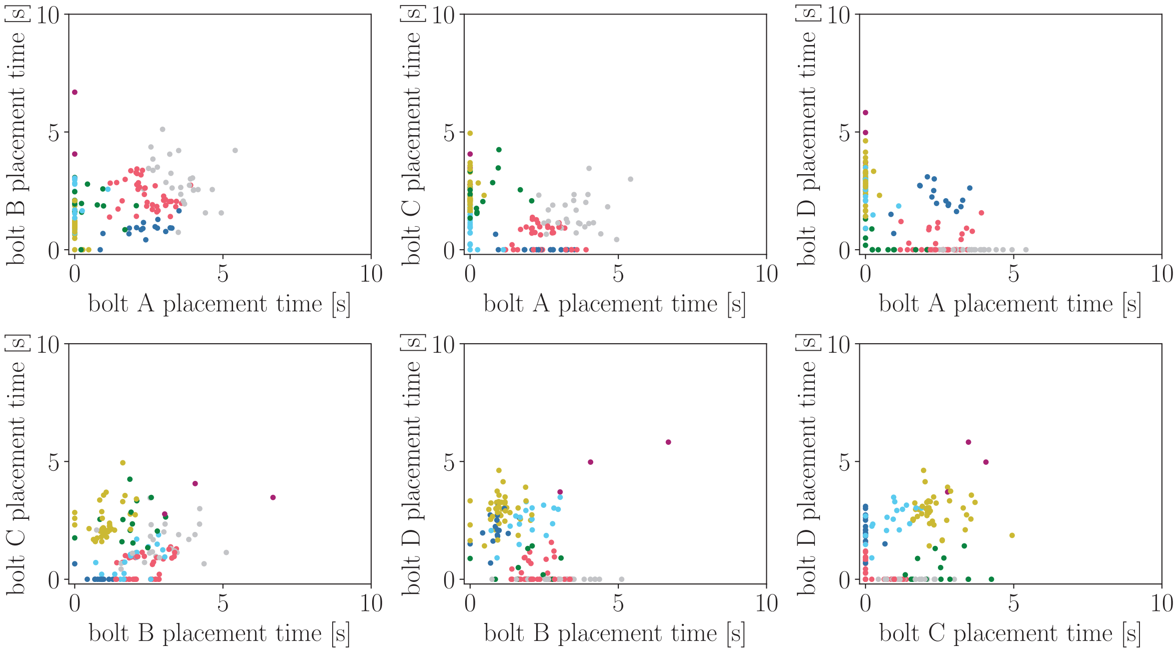


Figure 6.22: K-means clusters in the assembly data belonging to operator three of the bolts for all blocks, depicted using the blocks' bolt indices

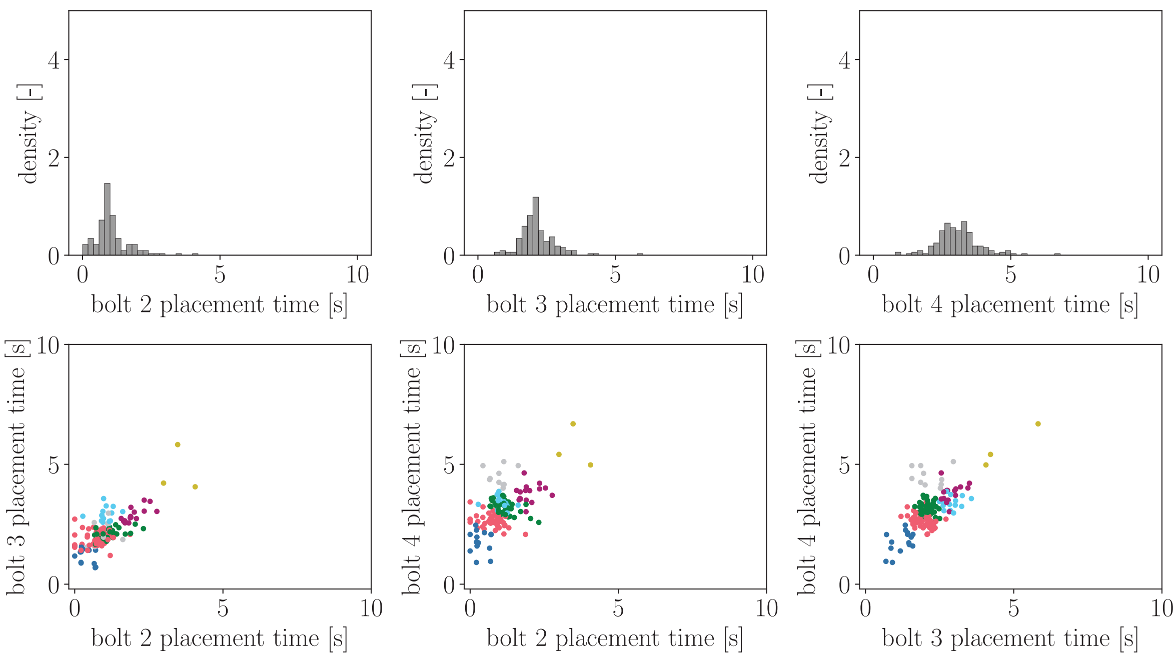


Figure 6.23: K-means clusters in the assembly data belonging to operator three of the bolts for all blocks, depicted using the blocks' bolt placement ranking

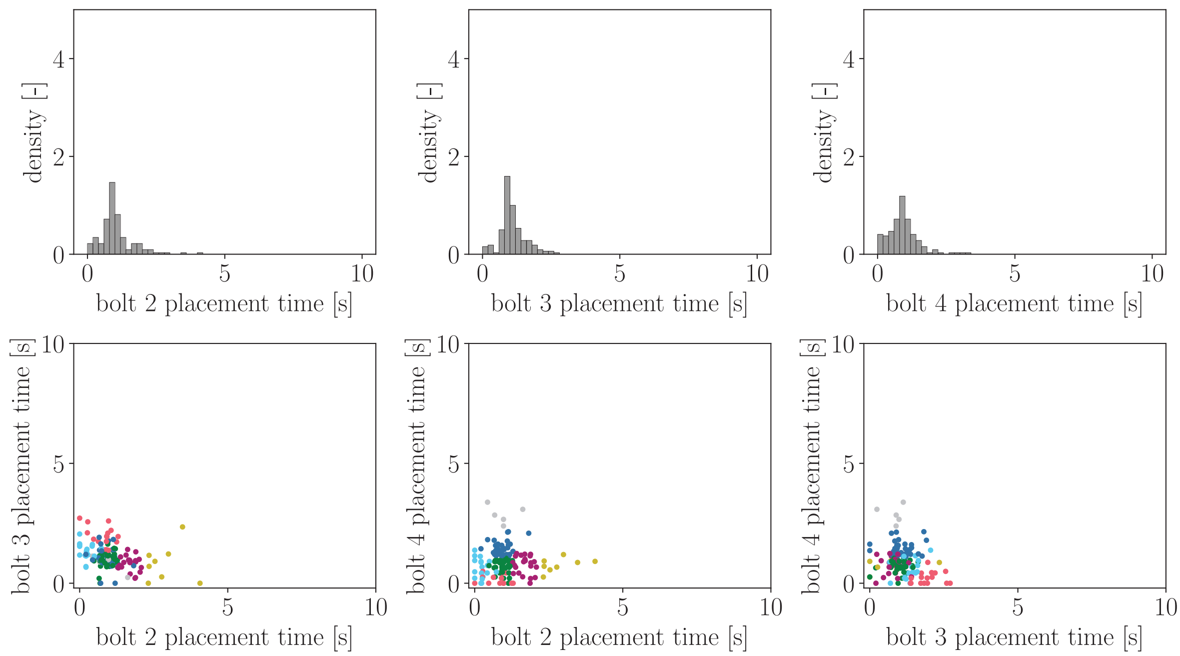


Figure 6.24: K-means clusters in the assembly data belonging to operator three of the bolts for all blocks, depicted using the blocks' bolt placement ranking in relative time

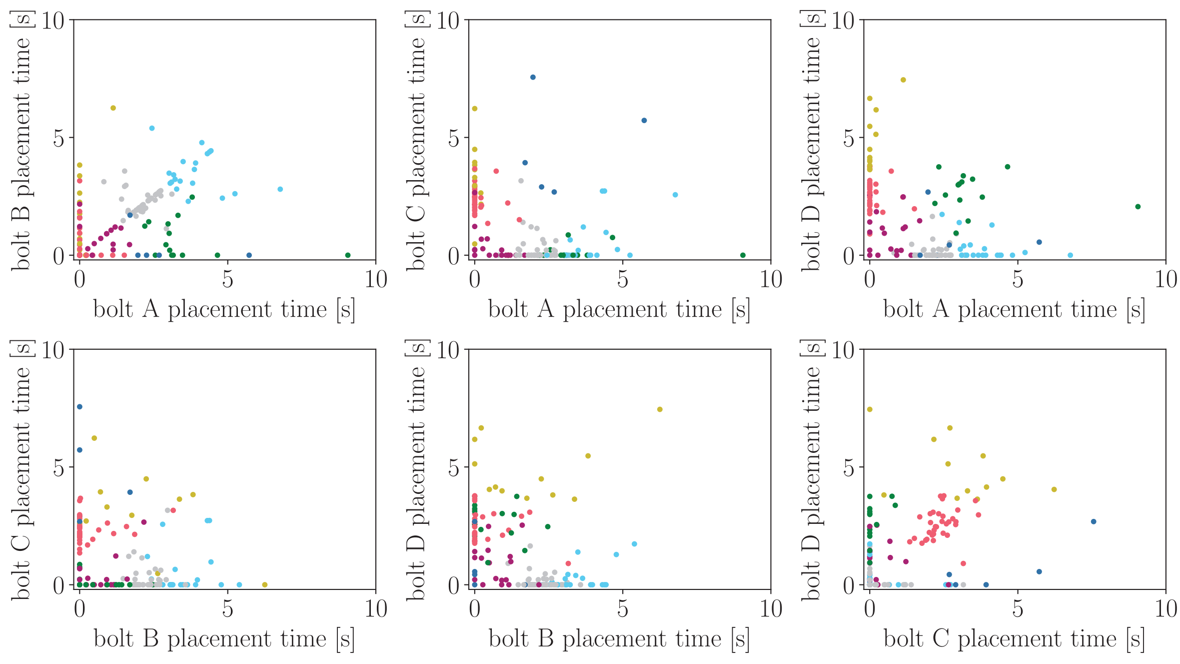


Figure 6.25: K-means clusters in the assembly data belonging to operator five of the bolts for all blocks, depicted using the blocks' bolt indices

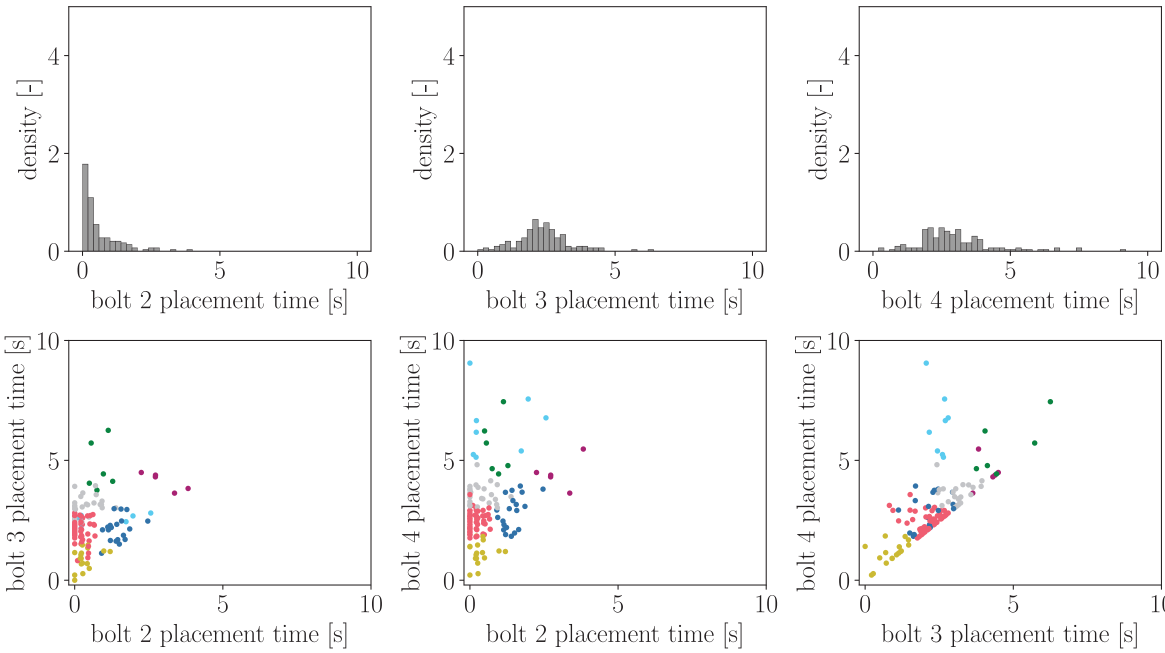


Figure 6.26: K-means clusters in the assembly data belonging to operator five of the bolts for all blocks, depicted using the blocks' bolt placement ranking

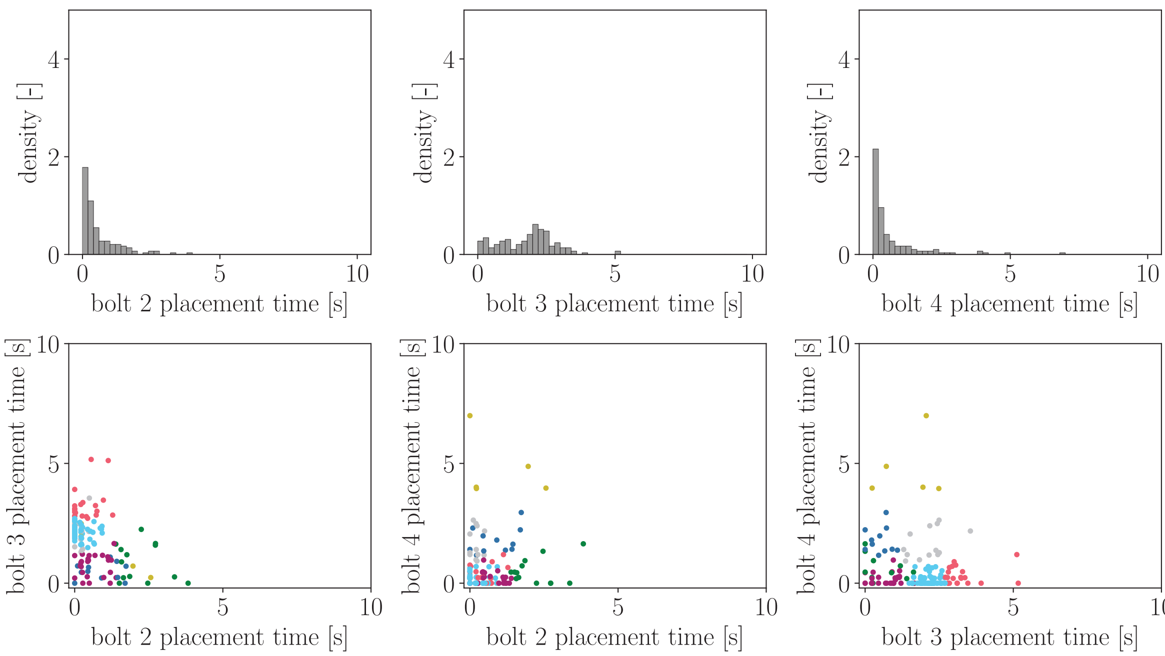


Figure 6.27: K-means clusters in the assembly data belonging to operator five of the bolts for all blocks, depicted using the blocks' bolt placement ranking in relative time

It is expected that the clustering finds sets of bolts that have been placed at similar times, or at least a subset of the bolt indices at similar times with for example one index being of a different value. These groupings represent similar placement times of bolts within blocks, but are not linked to these blocks itself. The next step is therefore to depict the resulting clusters against the block features as these can

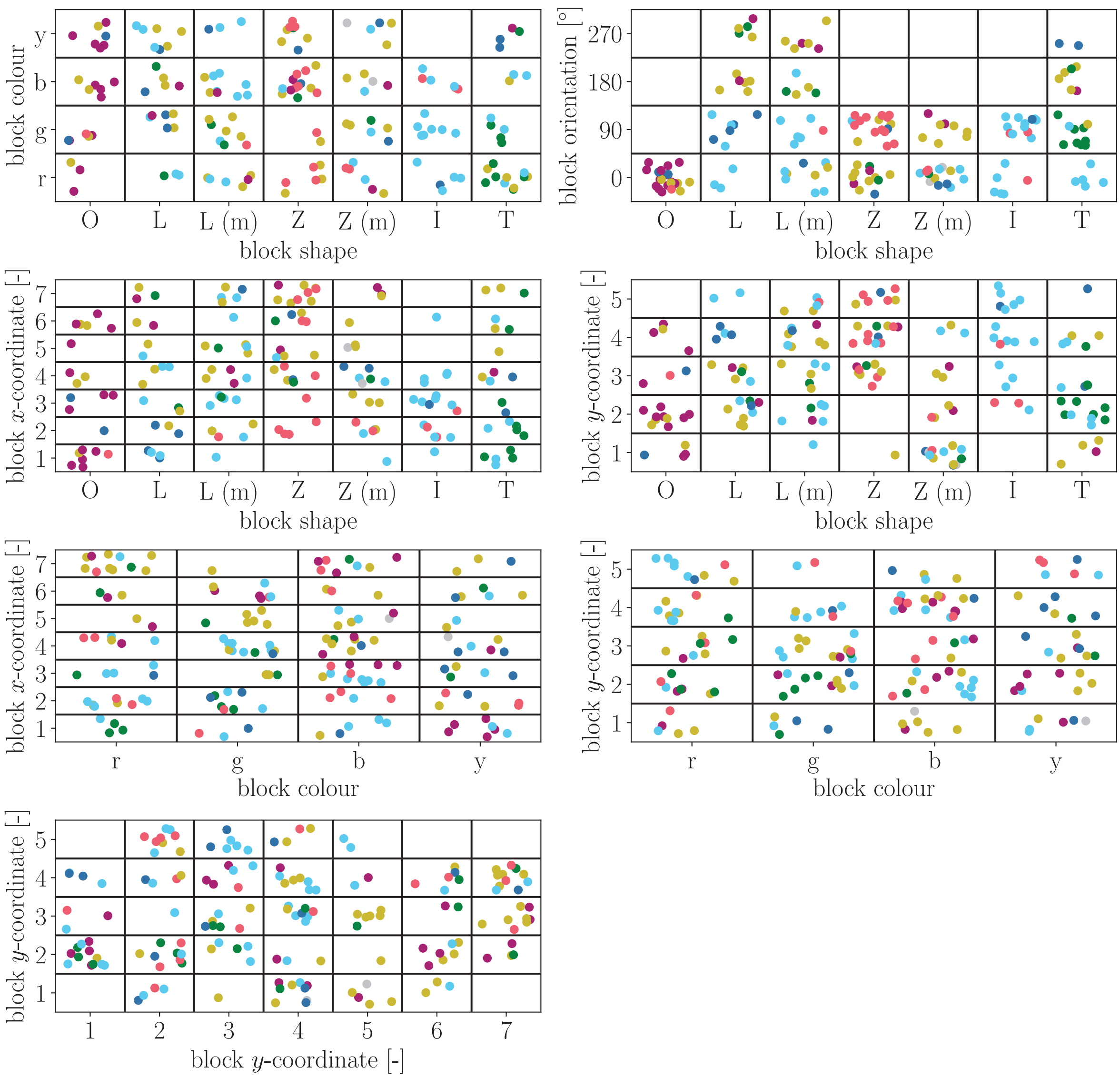


Figure 6.28: Resulting clusters of K-means clustering of assembly data operator three depicted against block features

indicate if the response of the operator is correlated to the block feature values. If this is the case, and the reason for faster or slower assembly of certain blocks can be found, assembly instructions can potentially be adapted to overcome the problem causing slower assemblies or to encourage the behaviour seen during faster assemblies.

The clustering has been applied to the response of the assembly, so are there relations to be found in the product and block features with consistent response clusters, so that future product assemblies probably also belong to that cluster? Figure 6.28 depicts the generated clusters versus feature values of the products for operator three, and Figure 6.29 for operator five. Within the squares that are defined by the intersection points of the features depicted on the horizontal axis and vertical axis, the clusters off all assembled blocks by the respective operator are depicted using the same colours as in the response clustering, as they represent the same data sample. The locations of the samples within the squares are randomly distributed to not give the false indication of some sort of correlation within feature values. If a relation between the assembly time and feature combination exists, the same cluster allocation is expected, or multiple as of course one cluster of the response data could be linked to a certain feature

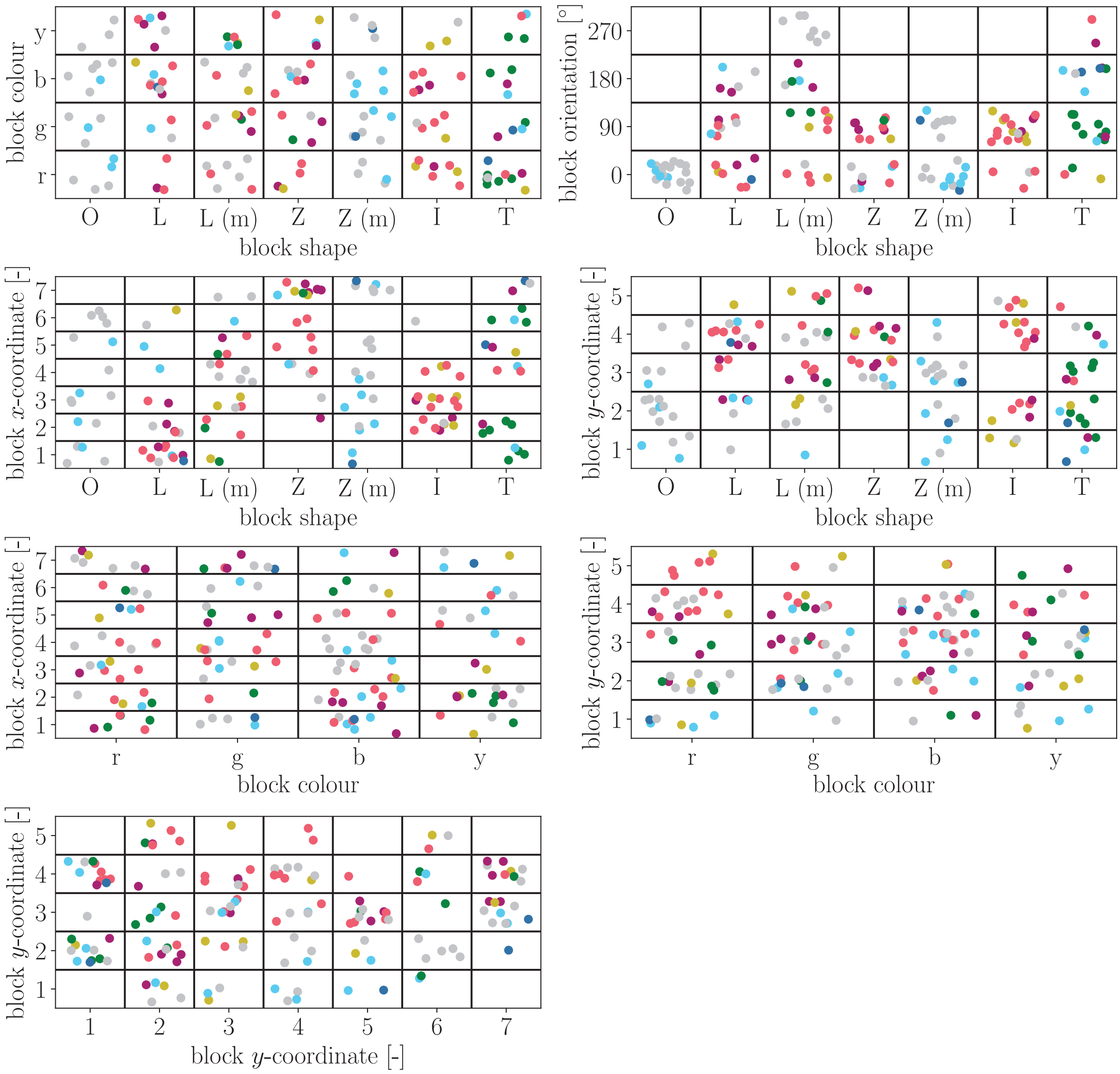


Figure 6.29: Resulting clusters of K-means clustering of assembly data operator five depicted against block features

value, another cluster to another feature value and another cluster maybe to multiple feature values. Based on previous results presented in this chapters, the found clusters are not expected to be dependent on the product features. However, the whole point of clustering is to overcome manual analysis limitations so next the results are discussed.

For operator three, the clusters indicated with purple markers ■, seem to be mostly clustered at the O-shape. An I-shape seems to be fairly constantly assembled within the cluster with light blue markers ■, but this cluster is not solely located at the I-shape, so could be caused by the I-shape just being assembled consistently. The Z-shape at an orientation of 90° is mostly assembled within the cluster with pink markers ■ and this cluster is mostly unique to this feature combination. For operator five, the O-shape and mirrored Z-shape seem to be assembled within two clusters, indicated with grey markers ■ and light blue markers ■. The grey markers are not unique to these shapes however. These results are limited, so no further clustering techniques are applied.

6.11 Conclusion

While unexpected results were found in the data analysis, none proved to be a research braking problem. However, the occlusion of the product platform with the undesired result of, potential parallel, bolt placement recognition times unequal to the real placement time should be fixed. Preventing the occlusions in the first place would be preferred, but by knowing when certain bolt positions are occluded, the occurrences of recognising bolt placements on times unequal to the real placement time can be at least filtered.

Assembly performance between operators clearly differs, making direct comparisons difficult. This is partially fixed by introducing scaled assembly times. A comparison solely based on the features versus assembly time did not result in conclusions, as features like block shape and block orientation are too dependent on each other. Analyses of two block features per operator did result statistically differences in assembly performance, but by fixing two feature values sample assembly occurrences are sometimes lacking. The analysis of the blocks' bolt indices per block shape versus the assembly performance shows clear assembly performance preferences, but also again showed the dependency of block orientation on the block shape. An analysis of the separate block orientations showed that operators have preferences in assembly order of the blocks, and also differ in way bolts are placed, namely in serial or parallel. A more extensive analysis of these local block strategies is performed in Chapter 8.

Operators indicated that they sometimes preferred certain orders in assembly, which is shown to be true for certain features like the block colour and x-coordinates and y-coordinates. The effect of this preference on assembly performance is recommended to look into in future research, as the existence of the preference is shown. The same cannot be said for colour dominance, products with multiple blocks in the same colour, for which no effect has been found on assembly time. Assembly improvement over time is observed slightly, but the learning curve is not steep. The variance on the assembly time data of the blocks is too high for clear conclusions concerning the existence of a learning curve for block shapes. Clustering of the assembly performance based on the block indices did not yield correlations to block features, and is not pursued further.

The next chapter covers the strategies operators use to assemble products, and their impact on performance. If it can be shown that certain strategies are beneficial for performance, operators can be directed to assemble following these strategies to increase efficiency.

Chapter 7

Global Strategies

As introduced in Chapter 6, they way operators assemble can be described with assembly strategies. A distinction between *global* strategies and *local* strategies is made. Global strategies describe the complete assembly of the product on block and bolt level, and local strategies describe the assembly within blocks solely on bolt level. Local strategies are further explored in Chapter 8. By describing operator assembly preferences in strategies, a relatively small set of features based on operator actions can be linked to assembly sequences which can be applied to all variants of products, if correctly defined. Possibly, following these strategies fully or partially can be correlated to performance. If strategies can be found which are positively correlated to assembly performance, these strategies can be applied to the assembly instructions of new products, which is expected to improve operator performance.

7.1 Introduction to Assembly Strategies

Strategies describe the assembly of the products in a summarising and overlapping manner, which makes it possible to get bigger sample sizes to perform a statistical analysis concerning the way operators assemble products. For example, all product variants in this research can be assembled in a way that colours are assembled sequentially, while all products can still be different. As introduced, this opens up possibilities of finding correlations between these strategies and the performance of operators. If these correlations exist, adapting the assembly instructions has the potential of improving the assembly performance, as the positively correlated strategies can then be instructed to be worked with. Here, positively correlated means more correlation to a strategy that is positively correlated to performance (negatively correlated with assembly time). First, theoretical strategies operators could potentially follow are introduced based upon feedback received during the experiments, as introduced in Chapter 5. When the theoretical strategies are defined, they can be checked against how well operators follow them. If this can be shown, the amount of correlation of an assembly sequence to a strategy can be correlated too, namely to performance, which would show how much the following of a strategy in a certain strictness effects assembly performance.

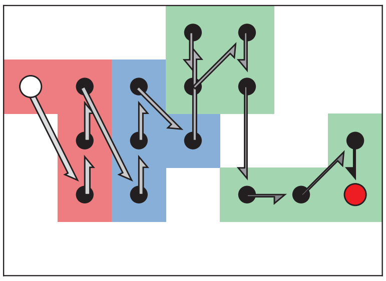
7.2 Theoretical Global Strategies

Based on operator feedback, a set of theoretical global strategies is defined that describe the way operators have been observed to assemble the products. The big drawback of this approach is therefore the necessity of defining the strategies by hand, which has multiple downsides. Firstly, it is operation dependent, which makes the strategies not easily translatable to other setups, and by definition limited by the imagination and knowledge of the programmer. Secondly, if assemblies cannot be matched to theoretical strategies, apparently some kind of strategy was not thought of, assuming the operator did follow a strategy. It should however be noted that operators did indicate that they were just doing something during assembly due to the simplicity of the task, so random assembly behaviour can be expected which is not a strategy that can be defined in a way that the actual assembly order could be correlated to this theoretical random order assembly strategy. The following is a list of the defined theoretical global strategies, with sub-lists indicating the variants of the overall strategies:

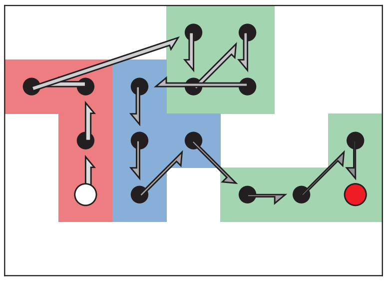
- left to right directional assembly (LR);
- left to right and top to bottom bidirectional assembly (LR-TB);
- left to right and bottom to top bidirectional assembly (LR-BT);

- right to left directional assembly (RL);
- right to left and top to bottom bidirectional assembly (RL-TB);
- right to left and bottom to top bidirectional assembly (RL-BT);
- top to bottom directional assembly (TB);
- bottom to top directional assembly (BT);
- block by block left to right assembly (LR);
- block by block right to left assembly (RL);
- block by block top to bottom assembly (TB);
- block by block bottom to top assembly (BT);
- block by block assembly without directional specification;
- colour by colour assembly;
  - note that this strategy has 24 permutations as there are four colours present in the products;
- assembly of two bolts at once.

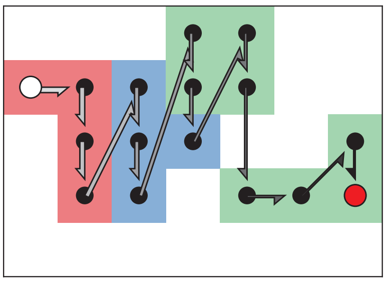
In summary, directional, bidirectional, colour based, directional block based and strategies based on parallel placement are defined. Four examples of strategies are depicted in Figure 7.1, with marker ○ indicating the beginning of the assembly and marker ● indicating the end of the assembly. Arrows indicate the assembly order, with the arrow colour linearly transforming from light grey to black based on the placement times of the corresponding bolt. The visualisation of the platform and the bolts is simplified by omitting the unused bolt holes in the platform and removing the formatting of the platform and bolts. The colours of the blocks and bolts are indicated with background colours in the respective block's colour. Now that theoretical strategies are defined, they have to be matched to the actual assembly orders. This *strategy maching* is introduced next.



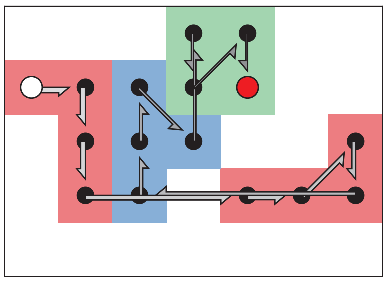
(a) Example of assembly from left to right



(b) Example of assembly block by block



(c) Example of assembly from left to right and top to bottom



(d) Example of assembly colour by colour and in this case also block by block

Figure 7.1: Examples of assembly strategies with marker ○ indicating the beginning of the assembly and marker ● indicating the end of the assembly

7.3 Strategy Matching

To match the theoretical strategies to the actual placement order of the bolts, both are represented by a list of the bolt indices in placement order of the bolts. While some strategies like working from left to right and top to bottom define a unique placement order for a match of the strategy to the actual placement order, other strategies like block by block define non-unique placement orders as different orders of placement could still represent following a certain strategy strictly.

7.3.1 Permutations in Theoretical Strategies

These possible non-unique placement orders should all be taken into account while checking if the assembly has a strategy match. As an example consider the theoretical strategy of colour by colour assembly applied on product $P_{7.2}$ as depicted in Figure 7.2. Parts of the corresponding theoretical assembly order based on the strategy may be in arbitrary order, as the within colour bolt order does not matter and the sequential order of the colours itself neither. These arbitrary orders potentially create big numbers of permutations, which is computationally undesired. Colour by colour assembly of product $P_{7.2}$ can be done in almost $3! \cdot (4! \cdot 4! \cdot 8!) = 1.4 \cdot 10^8$ ways. However, as the best match within the permutations is desired, actually calculating the possible permutations is not needed. Only one theoretical order for every actual placement order needs to be checked, which is worked out in section 7.4.2. First consider the theoretical bolt order using bolt indices in an ordered list, as given by

$$[[\mathbf{1},\mathbf{5},\mathbf{6},\mathbf{7}],[\mathbf{10},\mathbf{11},\mathbf{12},\mathbf{16}],[\mathbf{14},\mathbf{15},19,\mathbf{20},\mathbf{22},\mathbf{27},\mathbf{30},\mathbf{31}]], \tag{7.1}$$

where sub-lists of the full list are indicated by brackets. The indices are as introduced in section 2.2. This example depicts assembly per colour, specifically red-blue-green. The content of the sub-lists may be in arbitrary order for the match to be correct. In this case the sub-lists itself may be in arbitrary order too as all permutations resemble assembly colour by colour. Therefore for the sake of this example, a match is sought to the specific colour by colour strategy red-blue-green, which fixes the inter sub-list order. Now that it is known how to theoretically deal with permutations in the theoretical strategies, the next step is to deal with permutations in the actual bolt placement orders.

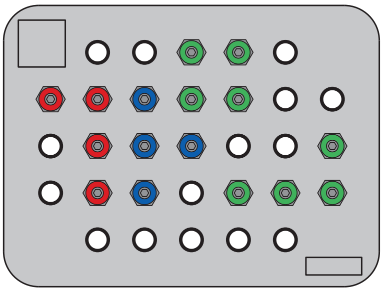


Figure 7.2: Layout product $P_{7.2}$

7.3.2 Permutations in Actual Bolt Placement Orders

Non-unique actual placement orders occur when two or more sequential bolts are placed at the same time. As a strict ordering cannot be defined here, permutations are generated for all possible strict orderings of the placement order. This also occurs when bolt placements are not recognised when placed due to occlusions of the camera. As the placement of more than two bolts at the same time is assumed to be not physically possible and therefore part of a not valid assembly, as introduced in section 6.1.3, such assemblies are not checked. Therefore, the computational worst-case scenario is placing two bolts at the same time for eight times, which gives a maximum of $(2!)^8 = 256$ permutations, which is computationally feasible to check completely. An assembly with parallel bolt placement is visualised in Figure 7.3. Bolts that are placed at the same time are connected with a coloured line coloured by its corresponding bolts, and the arrows connect to the midpoint of these lines to emphasise the non-uniqueness of the assembly. Note that all possible unique orders of a non-unique assembly are checked because the strategies are defined without taking parallel placement in account, with the exception of the strategy of placing two bolts at a time.

Now theoretical and actual strategies can be matched in a way that the strategy is used in the complete assembly without deviations. However, the correlation of the actual strategy to the theoretical strategies is also of importance in case full matches are not found while imperfectly following the strategies still could be correlated to assembly performance. Therefore, correlation of actual and theoretical placement orders based on strategies is explored next.

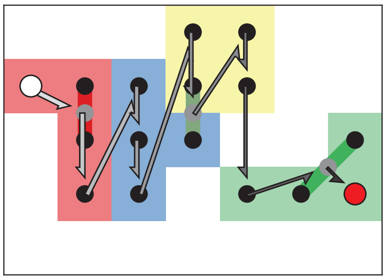


Figure 7.3: Assembly order product $P_{7.3}$ with marker ○ indicating the beginning of the assembly and marker ● indicating the end of the assembly

7.4 Correlation of Theoretical and Actual Placement Order

Until now, only matching the theoretical strategies with the actual strategies based on the placement order of the bolts is discussed. However not only exact matches are of interest, but also the closeness of non-matches to a match. To implement this, Kendalls τ correlation coefficient is used to define a measure of correlation between the actual and theoretical placement orders based on strategies [45].

7.4.1 Kendalls Tau Correlation Coefficient

Kendalls τ correlation coefficient compares the number of concordant and discordant pairs of two rankings. Two pairs are concordant if both pairs are increasing or decreasing. A pair is discordant if one of the pairs is decreasing and the other is increasing. As all pairs are compared, a swap of two close by pairs has a smaller negative impact on the correlation score than a pair that is not close by. An explanatory example of this is given in section 7.4.3. However, both the theoretical and the actual placement orders corresponding to strategies can be non-unique, how to calculate the correct correlation in this case? As in the given example of the colour by colour strategy, the order of bolts of a certain colour still does not matter, as long as all bolts of that colour are placed without placing other coloured bolts in between. As the closeness of a match is sought after, *ranking* optimisation of the theoretical bolt orders is performed, which is introduced next.

7.4.2 Ranking Optimisation

The theoretical bolt order using bolt indices of the colour by colour strategy red-blue-green of product $P_{7.2}$ is given by (7.1), with the bolts sorted by bolt index within the sub-lists. A fictive bolt placement order is given by

$$[\mathbf{7}, \mathbf{6}, \mathbf{5}, \mathbf{1}, \mathbf{12}, \mathbf{11}, \mathbf{16}, \mathbf{10}, \mathbf{15}, \mathbf{20}, \mathbf{14}, \mathbf{19}, \mathbf{22}, \mathbf{27}, \mathbf{31}, \mathbf{30}], \tag{7.2}$$

which logically gives the bolt ranking order given by

$$[\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{12}, \mathbf{13}, \mathbf{14}, \mathbf{15}, \mathbf{16}]. \tag{7.3}$$

The ranking order represents the placement position of the corresponding bolt in the same position in the actual bolt order in (7.2). If the bolt indices of the theoretical placement order are substituted with the placement rank of the actual bolts indices and are sorted in increasing order within the allowed permutable (sub-)lists, the correlation is optimised. The orders are linked by the bolt indices, making these substitutions possible. As the ranking should match as well as possible, the permutation that is created by sorting in increasing order gives the best match. Therefore, all sub-lists only have to be sorted in increasing order. Substituting the bolt indices in (7.1) with the ranking in (7.3) linked using the corresponding bolt index in (7.2), gives a theoretical ranking order based on bolt indices given by

$$[[\mathbf{4}, \mathbf{3}, \mathbf{2}, \mathbf{1}], [7, \mathbf{6}, \mathbf{8}, \mathbf{5}], [\mathbf{11}, \mathbf{9}, \mathbf{12}, \mathbf{12}, \mathbf{13}, \mathbf{14}, \mathbf{15}, \mathbf{16}]]. \tag{7.4}$$

Sorting (7.4) chronologically within the sub-lists and loosing the sub-lists gives

$$[\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{4}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{12}, \mathbf{13}, \mathbf{14}, \mathbf{15}, \mathbf{16}], \tag{7.5}$$

which is a full match. Note that in the case of permutations with the actual placement order, all permutations should be checked and the highest correlation value should be chosen. As the correlations are optimised to be as high as possible, only positive values of the correlation will be depicted in visualisations of the correlation results. The correlation can have a value between -1 and 1, respectively meaning exact reverse correlation and exact correlation. A value of zero means no correlation is present. In summary, these ranking orders are generated by taking the theoretical order based on the bolt indices, and translating them into the corresponding ranking of the placed bolts by linking the bolt indices of the ordered actual bolt placement indices.

Table 7.1: Example orders, with *th.* short for *theoretical order*

index	actual	th. 1	th. 2
A	1	1	4
B	2	3	2
C	3	2	3
D	4	4	1

Consider the example actual and theoretical orders in Table 7.1, with the indices represented by letters for clarity. Comparing all pairs gives the concordant pairs (indicated with C) and discordant pairs (indicated with $\bar{C}$) as given in Table 7.2. It can be seen that theoretical order one matches better to the actual order, and therefore also has a higher value for τ. The τ value is given by

$$\tau = \frac{C - \bar{C}}{C + \bar{C}}. \tag{7.6}$$

The expression for τ given by (7.6) is the simplified version of the expression for τ_b given in the documentation of SciPy, a Python package used for data processing [46].

Table 7.2: Overview of concordant and discordant pairs

pair	actual	th. 1	th. 2	actual vs th. 1	actual vs th. 2
AB	$1 < 2$	$1 < 3$	$4 > 2$	C	$\bar{C}$
AC	$1 < 3$	$1 < 2$	$4 > 3$	C	$\bar{C}$
AD	$1 < 4$	$1 < 4$	$4 > 1$	C	$\bar{C}$
BC	$2 < 3$	$3 > 2$	$2 < 3$	$\bar{C}$	C
BD	$2 < 4$	$3 < 4$	$2 > 1$	C	$\bar{C}$
CD	$3 < 4$	$2 < 4$	$3 > 1$	C	$\bar{C}$

7.4.3 Imperfect Correlation

The example of section 7.4.2 gives a perfect match, but the reason Kendalls τ is implemented, is to deal with imperfect matches. When the theoretical and actual strategies match, the correlation equals one. Consider the example theoretical ranking order as given by

$$[\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{5}, \mathbf{4}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{12}, \mathbf{13}, \mathbf{14}, \mathbf{15}, \mathbf{16}], \tag{7.7}$$

which gives a correlation coefficient of $\tau = 0.983$. This is desired, as the strategies almost match. Switching two bolts around should have a limited effect. However, now consider the following example of a theoretical ranking order as given by

$$[\mathbf{1}, \mathbf{2}, \mathbf{3}, \mathbf{13}, \mathbf{5}, \mathbf{6}, \mathbf{7}, \mathbf{8}, \mathbf{9}, \mathbf{10}, \mathbf{11}, \mathbf{12}, \mathbf{4}, \mathbf{14}, \mathbf{15}, \mathbf{16}], \tag{7.8}$$

which gives a correlation coefficient of $\tau = 0.715$. The fourth and 13$^{\text{th}}$ bolt are switched in ranking instead of the fourth and fifth. This lower correlation value can be considered desired as the difference in the order is way more significant than just switching around two sequential bolt placements. Note if for example the third and second bolt would be switched, that this still results in a correlation of 1 as the ranking optimisation would still result in (7.5).

Having defined the theoretical strategies, knowing how to deal with non-uniqueness in these strategies and in the actual placement orders and being able to correlate the theoretical and actual orders by use of Kendalls τ, the experiment data can be correlated to the defined theoretical strategies.

7.5 Correlation Results

Three different comparisons are performed. The first depicts the mean correlation scores per operator for every strategy to establish if operators follow and if so, differ or not in strategy following between them. The second is based on a comparison between the fastest and slowest assembled products per operator to see if the strategies have an effect on performance. The third depicts the performance level for all strategies per operator against the correlation score to give a complete overview of the correlation between the strategy correlation and performance. In these comparisons, it is again important to differentiate between the absolute assembly times and the standardised assembly times, as otherwise the fastest assembly strategies of the slowest overall operator will not show up in a comparison of the fastest assembly times, while this data could still be important.

For an overview per operator, the correlation scores are averaged for every strategy in Figure 7.4 for all block products assembled. Operators are indicated using the colour scheme introduced in section 6.4. The abbreviation R stands for right, L for left, T for top and B for bottom. It is expected that block by block assembly is the most common assembly strategy as this strategy was indicated by most operators as the desired strategy. Colour by colour is expected to be in close proximity, as block by block assembly also means assembling colour by colour to some extend. The difference is found in occurrences of multiple blocks in the same colour which are not assembled back to back. Furthermore, left to right assembly is expected to be more common than right to left due to cultural habits. In addition, operator five is expected to have a high score on two by two placement of bolts, as has been observed in section 6.6.2.

From the data it is clear that operators work by colour and by block, even for the later operators from operator eight onward who were not indicated to do so anymore. In addition, assembling by block and top to bottom seems popular. Working from left to right by bolt and by block are both more popular than their right to left variants. Concerning points of interest, operator twelve in colour ■ is more directional strategy wise, which is noticeable in the operators score for working by block. In addition, operator eight in colour ■ clearly works left to right by block, and operator six in colour ■ works top to bottom by block. Operator five in colour ■ indeed places bolts two by two often, and the others considerably less.

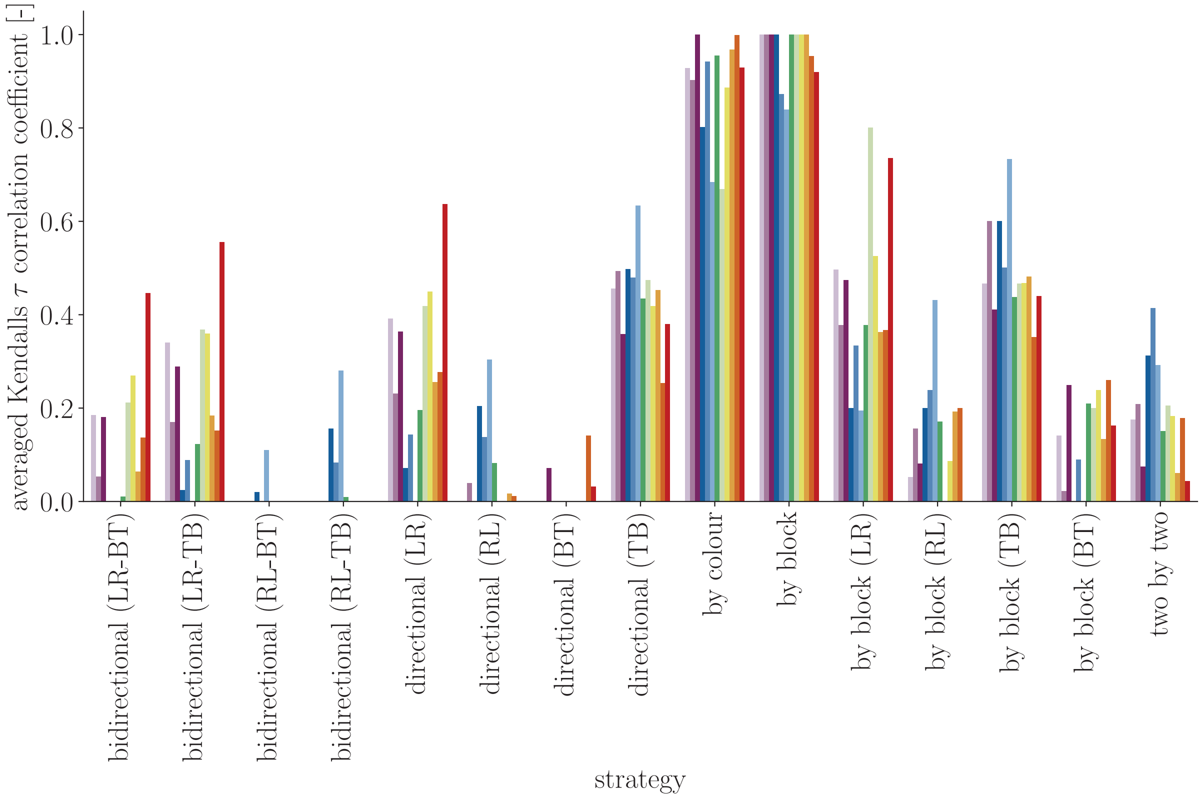
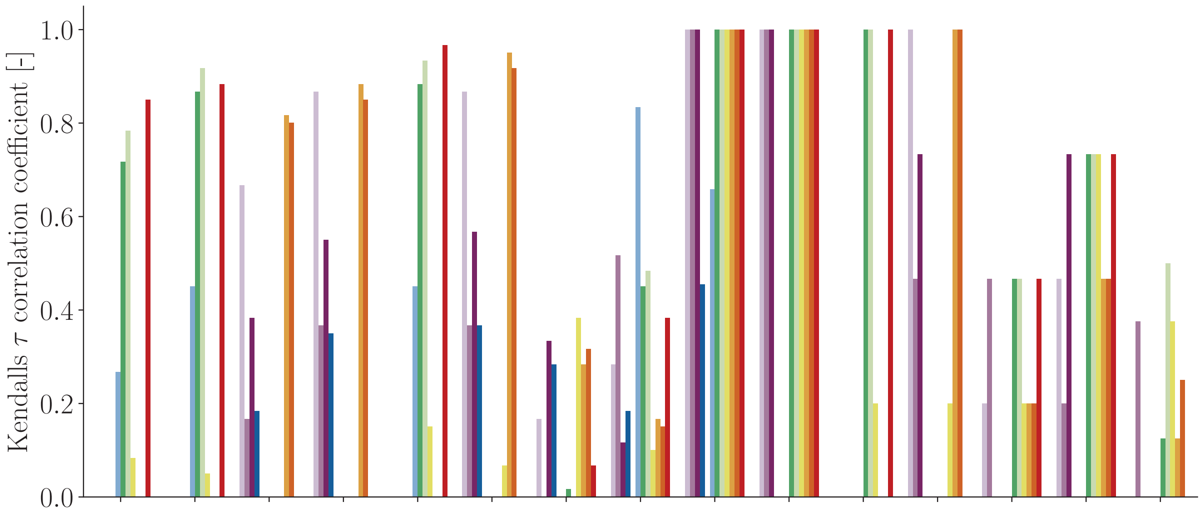
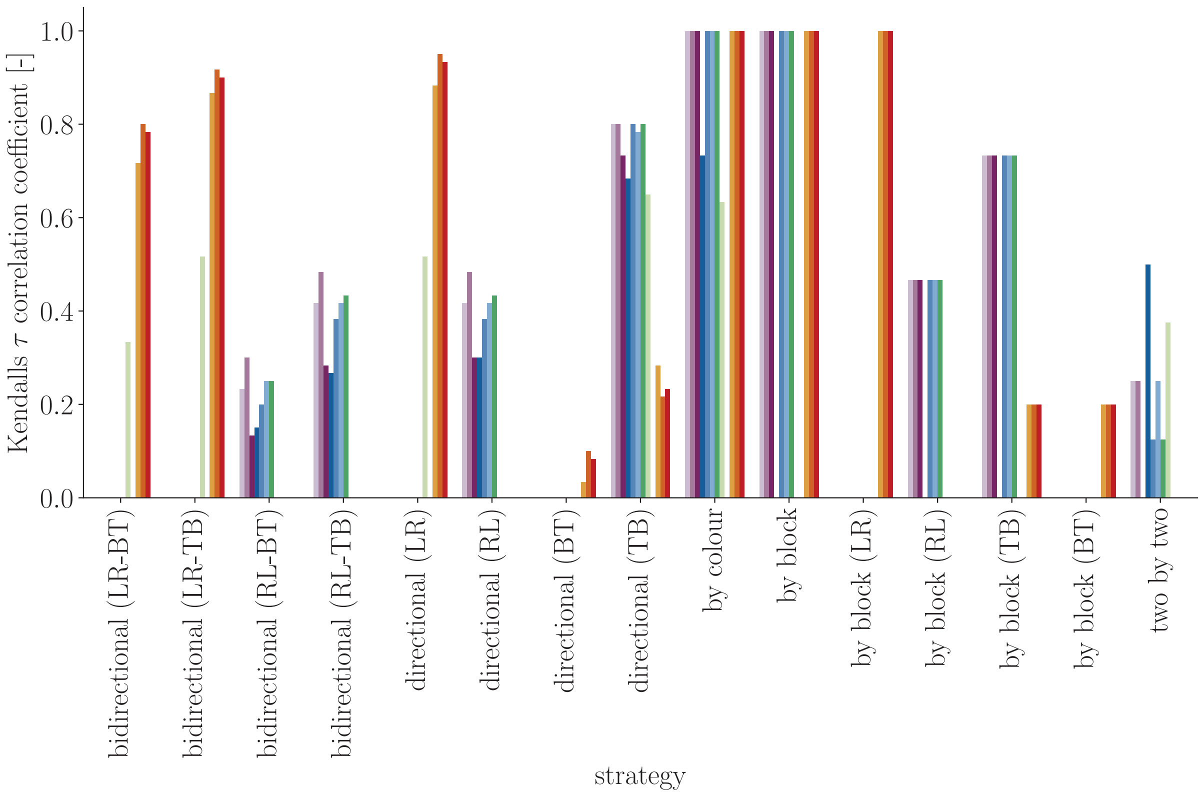


Figure 7.4: Averaged strategy correlation for every operator for every strategy, hues indicate operators in increasing order from left to right

Looking into specific products, Figure 7.5a depicts the strategies used to assemble product $P_{4.9a}$ depicted in Figure 4.9a, and Figure 7.5b depicts the strategies used to assemble product $P_{4.9b}$ depicted in Figure 4.9b. Note that due to recognition errors, data of certain product types is not always available. Product $P_{4.9a}$ consists out of four different colours, while product $P_{4.9b}$ only consists out of two colours that are also nicely split to the left and right of the product. Therefore, different strategies are expected for assembly, with notably for product $P_{4.9b}$ a more colour based assembly approach and therefore a more directional one too. For completeness, the assembly sequences the correlation coefficient values are based upon, are depicted in Figure 7.6.

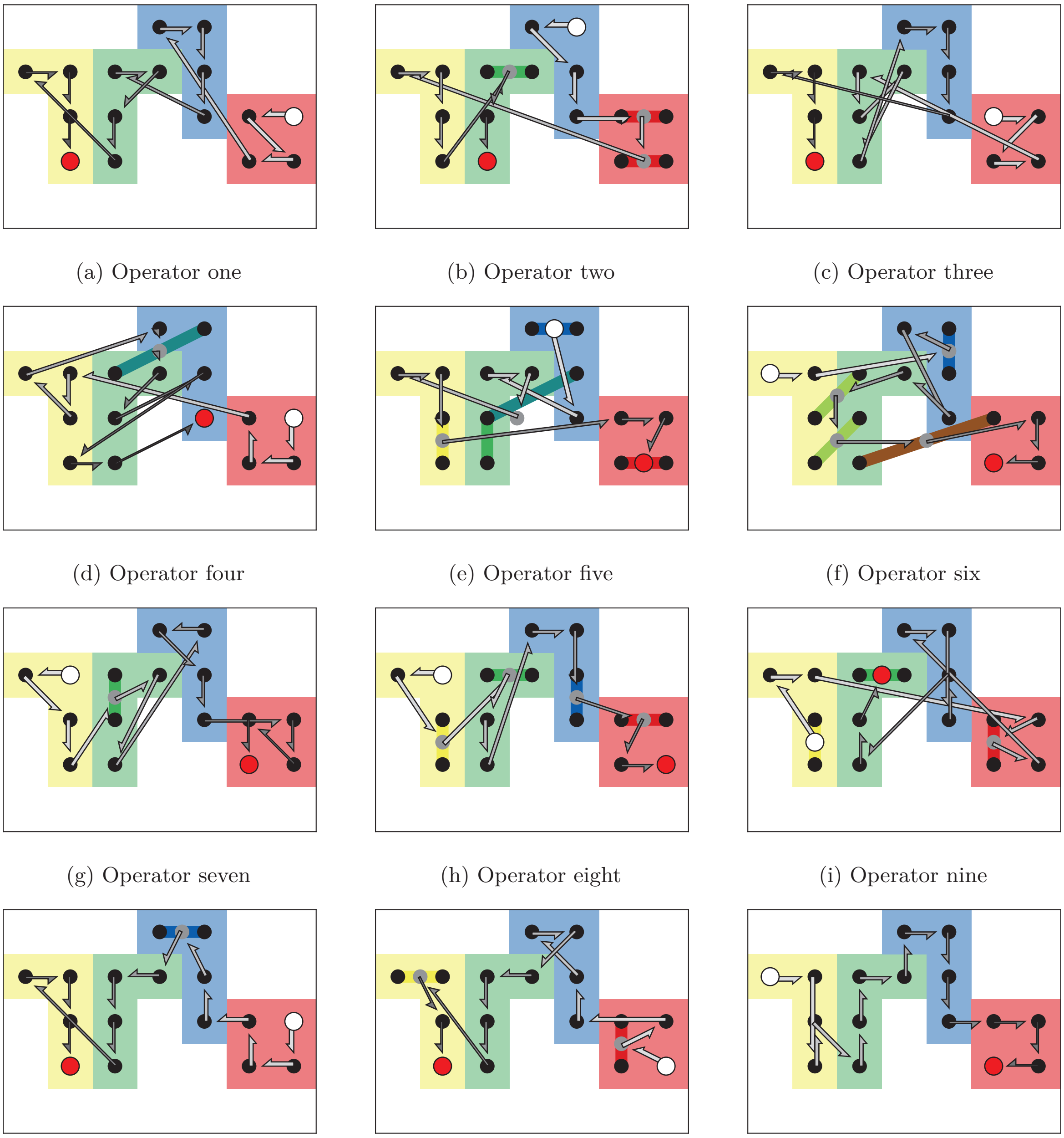


(a) Strategy correlation product $P_{4.9a}$



(b) Strategy correlation product $P_{4.9b}$

Figure 7.5: Strategy correlation products $P_{4.9a}$ and $P_{4.9b}$ for every operator for every strategy, hues indicate operators in increasing order from left to right



(a) Operator one (b) Operator two (c) Operator three

(d) Operator four (e) Operator five (f) Operator six

(g) Operator seven (h) Operator eight (i) Operator nine

(j) Operator ten (k) Operator eleven (l) Operator twelve

Figure 7.6: Assembly sequences of product $P_{4.9a}$ for all operators, with arrows indicating assembly order

From the data, it is indeed clear these strategies are very different, which means that the averaged values in Figure 7.4 do not represent repeating behaviour. However, the expected behaviour differences are not present. Both products are clearly assembled by block and by colour, with product $P_{4.9a}$ also being directionally assembled block by block, but product $P_{4.9b}$ considerably less.

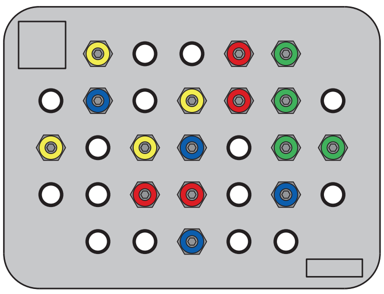
Looking at specific operators highlights, there are some clear differences. First note that there is no data of operator nine (■) for product $P_{4.9b}$. Operator seven (■) and eight (■) have high correlation for bidirectional left to right assembly of product $P_{4.9a}$, but barely no correlation for these same strategies for product $P_{4.9b}$. The same observations are made for left to right directional assembly. For assembly by block, no correlation at all is observed for these operators for product $P_{4.9b}$. Operators ten (■) and eleven (■) for product $P_{4.9a}$ prefer right to left based (bi)directional assembly, while for product $P_{4.9b}$

they prefer the left to right variants, with also a perfect correlation of working by block from left to right For operators four (■), five (■) and six (■) assembly by block and by colour are highly correlated for product $P_{4.9b}$ while for product $P_{4.9a}$ this is not the case. To a lesser extend, the same can be said for top to bottom and right to left assembly for product $P_{4.9b}$. For operators one (■), two (■) and three (■) however, the preferred assemblies seem quite similar and not significantly impacted by the differences in the products. For operator twelve (■), the same observation is made with the exception of directional assembly by block where the operator barely correlates to for product $P_{4.9b}$. Overall, clear differences between operators are observed, but there also operators who assemble quite consistently.

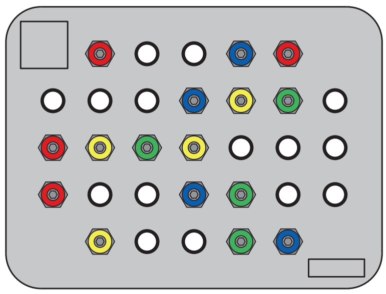
To check if operators changed strategy over time, comparing the first versus the last assembled products would desired, however as Figure 4.9a and Figure 4.9b showed that assembly strategies are not consistent over the products and due to the fact that the assembly orders for all operators were different, a comparison over time would not be fair to make as this comparison would be too dependent on the product types assembled.

Strategy correlation for random products is also analysed. Logically, block by block strategies cannot be applied, but the directional ones and the colour based ones certainly can. As it is now known that comparing a summary of the strategies is not representative of all products, two products from the assembly set are taken, product $P_{7.7a}$ as depicted in Figure 7.7a and $P_{7.7b}$ as depicted in Figure 7.7b. The results for product $P_{7.7a}$ are depicted in Figure 7.8a and the results for product $P_{7.7b}$ are depicted in Figure 7.8b.

More structure based assembly strategies are expected, with colour by colour being the most obvious one as directional based strategies would require constant switching of bolt colours. Indeed, both products are assembled colour by colour very strictly, with also some significant correlation for directionality for mostly operator nine (■) and ten (■) for product $P_{7.7a}$, but to a lesser extent for product $P_{7.7b}$. As no significant differences in strategy usage is found between operators, an comparison of the strategy correlation to performance of random based products is not performed.

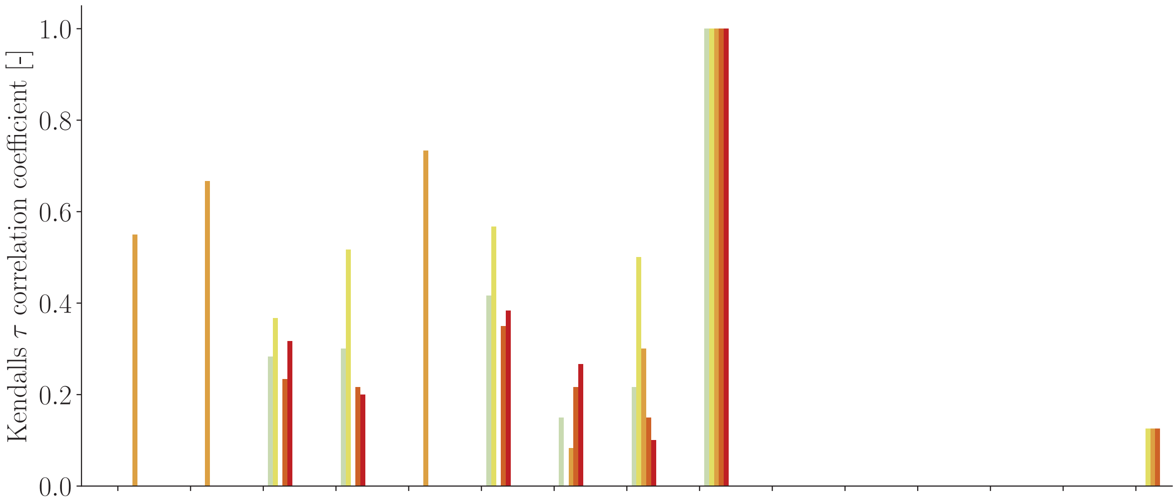


(a) Product $P_{7.7a}$

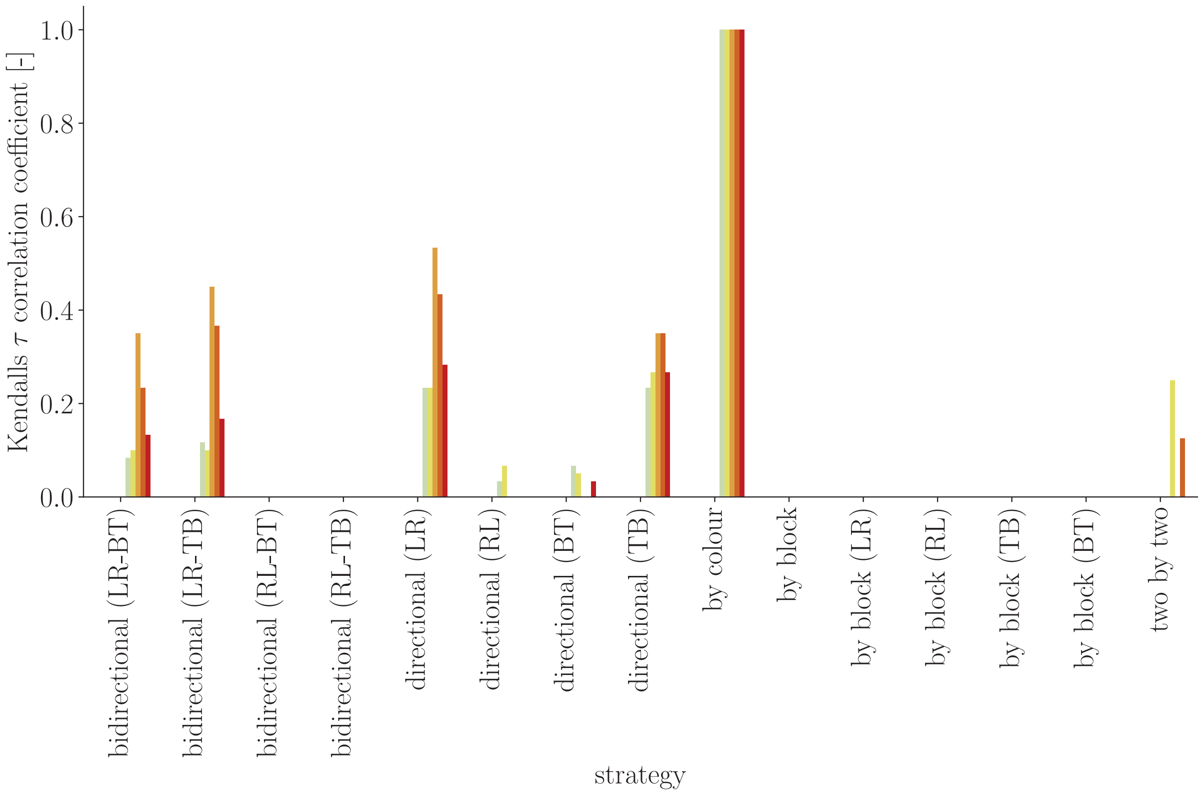


(b) Product $P_{7.7b}$

Figure 7.7: Two random based products



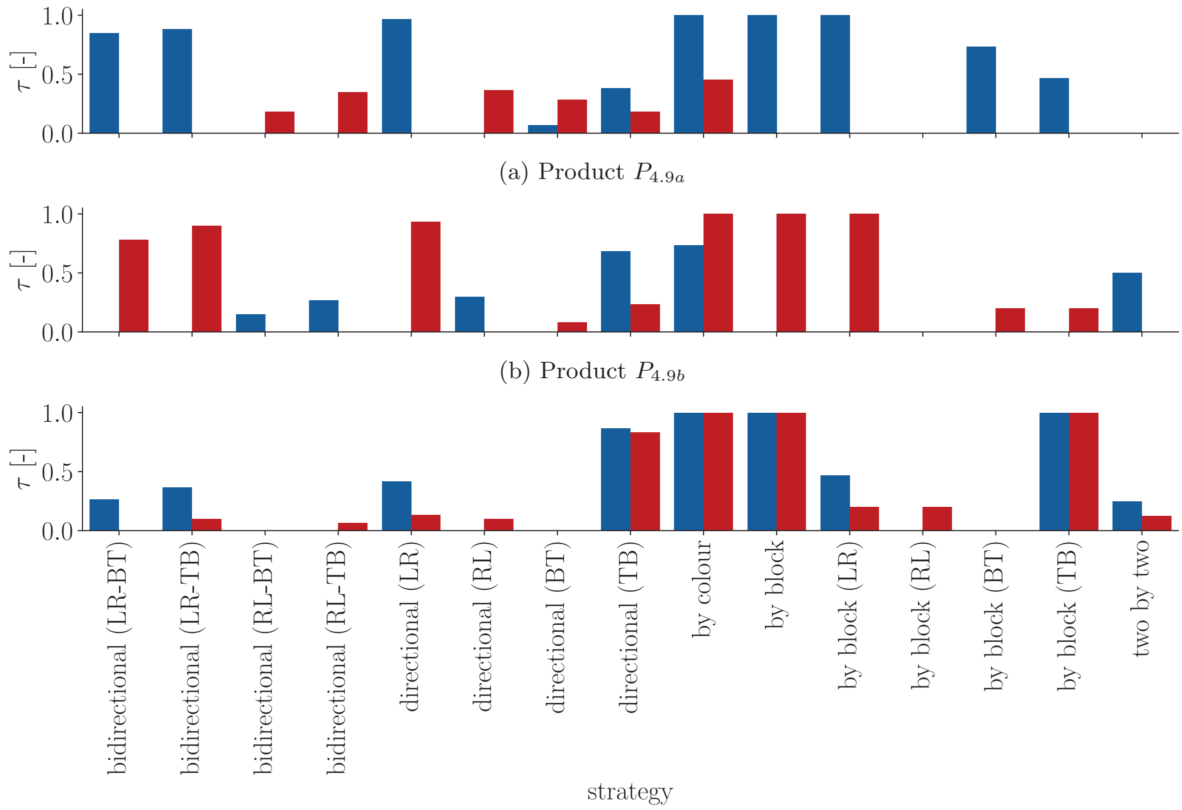
(a) Strategy correlation product $P_{7.7a}$



(b) Strategy correlation product $P_{7.7b}$

Figure 7.8: Strategy correlation product $P_{7.7b}$ and $P_{7.7b}$ for every operator for every strategy, hues indicate operators in increasing order from left to right

For products $P_{4.9a}$, $P_{4.9b}$, and $P_{4.9c}$, which are assembled by all operators, the relative slowest and fastest assembly realisations are depicted in Figure 7.9. The relative fastest and slowest assembly are chosen to not disadvantage and advantage the respectively slower and faster overall operators. If no value is depicted, the correlation is equal to zero or negative. As introduced, the highest positive value of correlations are taken by ordering the theoretical ranking orders, so negative values have no mathematical meaning as all introduced strategies have reversed counterparts. Ideally, a clear difference in strategies is observed for fast versus slow assemblies, which would indicate that certain strategies are more or less efficient. For all three products, the used strategies for both the fastest and slowest product differ between the products. Products $P_{4.9a}$ and $P_{4.9b}$ also differ for the fastest and slowest occurrence, while these are quite similar for product $P_{4.9c}$. Overall, strategy performance differs between products. As operators and products differ, a variant of this plot which averages assembly strategies over the users or products is not depicted.



(a) Product $P_{4.9a}$

(b) Product $P_{4.9b}$

(c) Product $P_{4.9c}$

Figure 7.9: Comparison of the relative fastest, in colour ■, and slowest assemblies of products, in colour ■

As operators indicated not always noticing repetition in products, a comparison of the strategy preferences for these repeating products is performed. Is there a difference between consistency between operators that did notice repetition, and those who did not notice the repetition? As every product that is repeated, is only repeated three times, an extensive analysis of performance improvement is not possible, however overall trends are checked manually and not found. The repetitions of product $P_{7.13a}$ by operator ten, who did notice repetition, are shown in Figure 7.10, the repetitions of product $P_{7.13b}$ by operator two, who did not notice repetition, are shown in Figure 7.11 and the repetitions of product $P_{7.13c}$ by operator three, who did not notice repetition, are shown in Figure 7.12. The products are respectively shown in Figure 7.13. Operator ten shows consistent assembly behaviour, as does operator two, with the exception of the first occurrence of the repetition. Operator three however shows significant different strategy occurrences per repetition. This was observed at more occurrences of repeating products for operators who did not recognise the repetition. As too few data is available to analyse performance effects, the effect of changing the assembly on performance is not known.

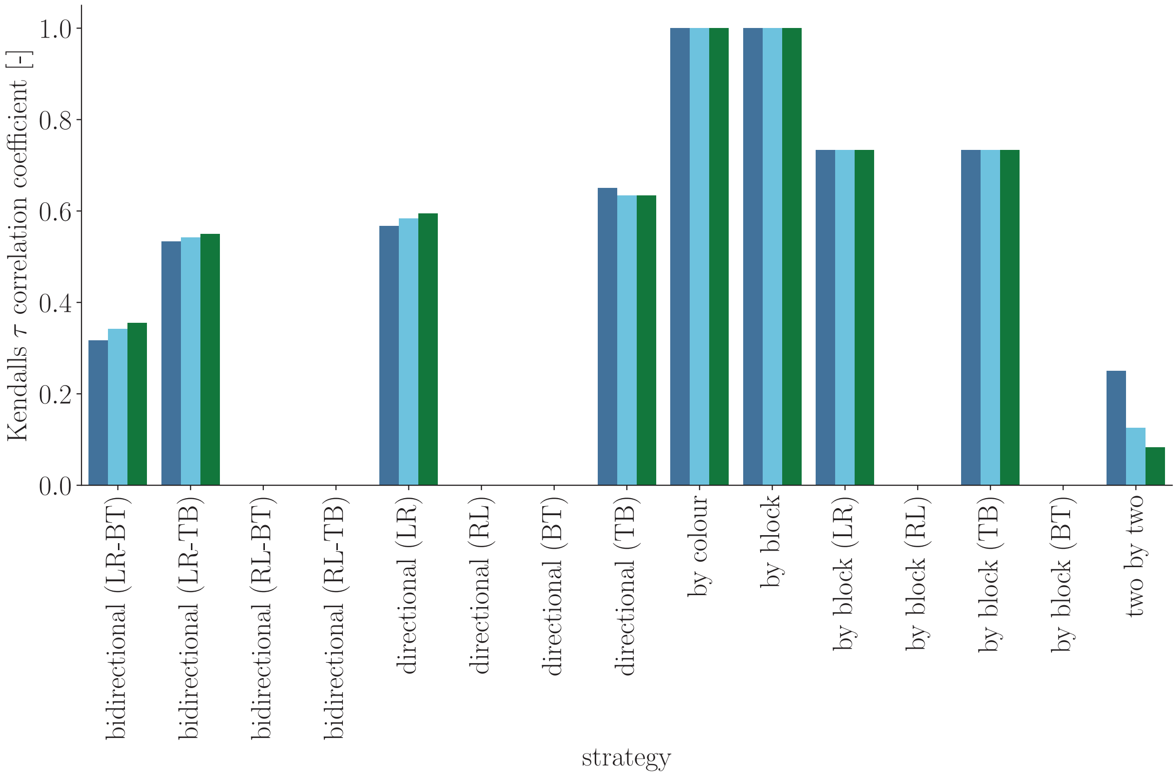


Figure 7.10: Strategy correlation for operator ten's repetitions of the same product

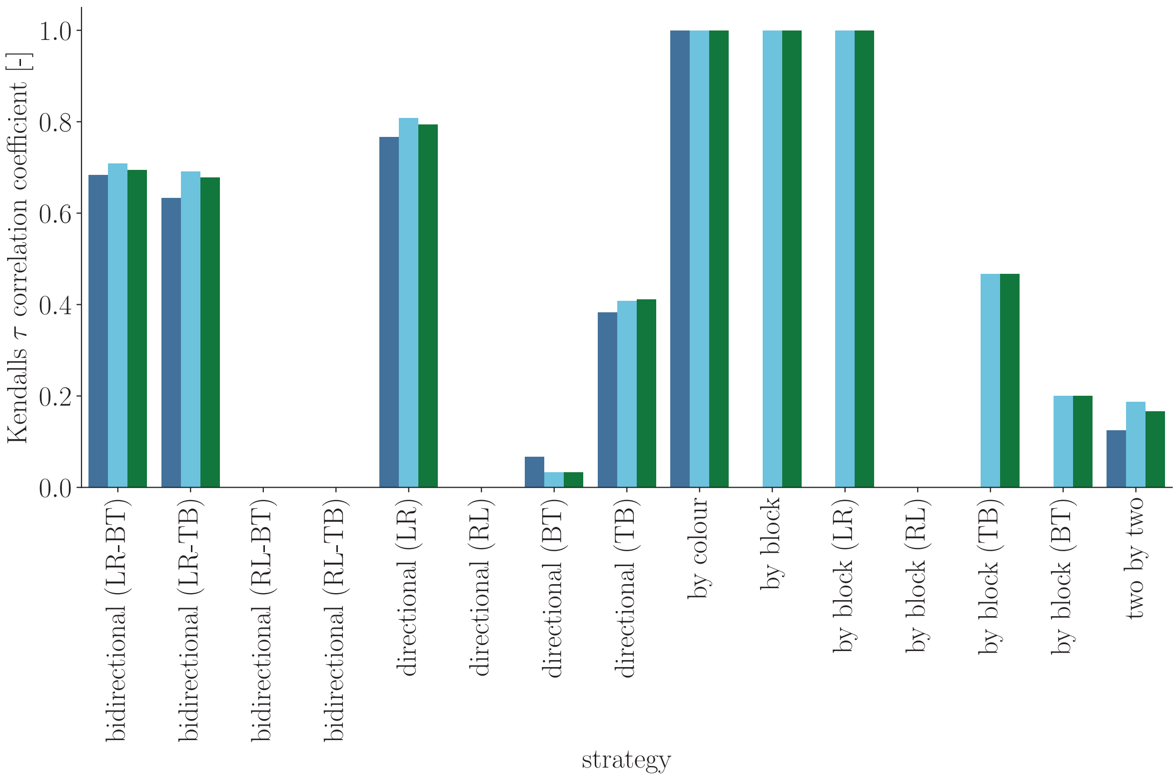


Figure 7.11: Strategy correlation for operator two's repetitions of the same product

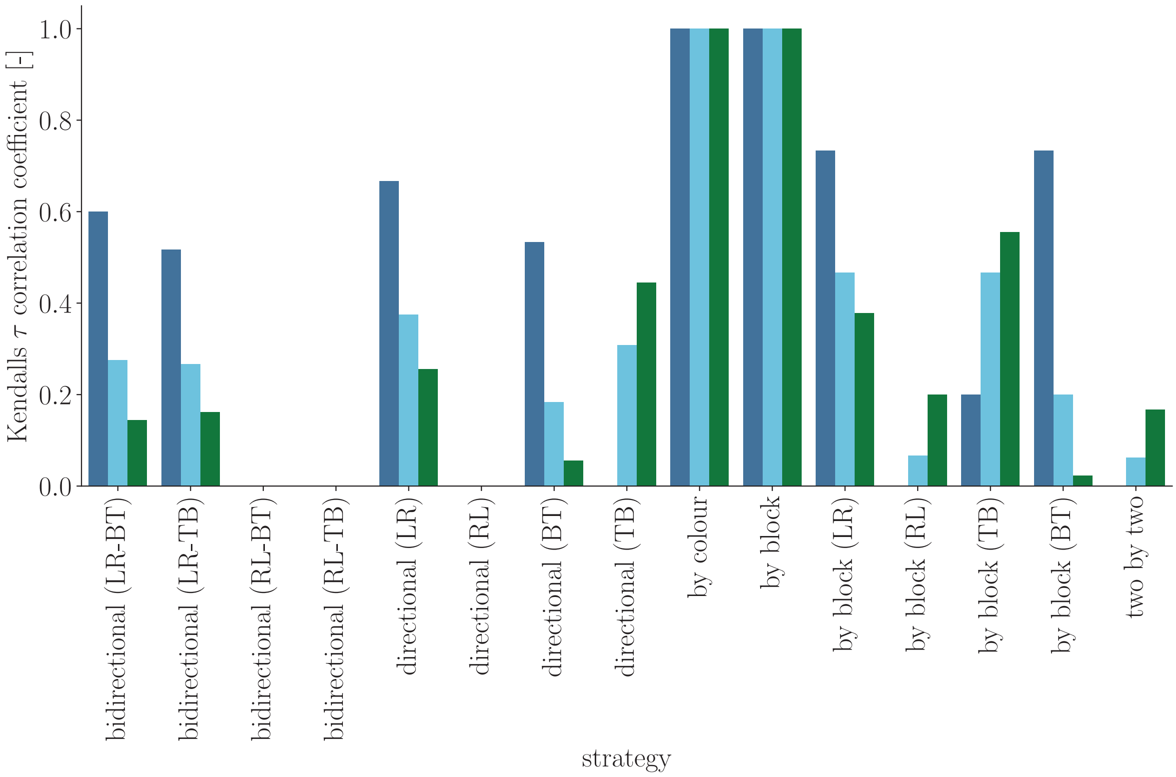
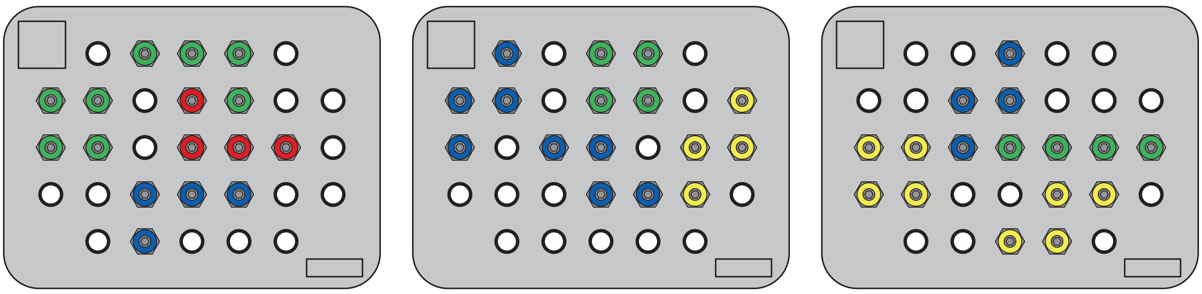


Figure 7.12: Strategy correlation for operator three's repetitions of the same product



(a) Product $P_{7.13a}$ (b) Product $P_{7.13b}$ (c) Product $P_{7.13c}$

Figure 7.13: Selection of products that are assembled multiple times by an operator

Figure 7.14, Figure 7.15, and Figure 7.16 depict the correlation of the strategy correlation to the operators scaled assembly time. For readability, the strategies are divided over three figures. The strategies are depicted by fixed colours, given by

- bidirectional (LR-BT);
- bidirectional (LR-TB);
- bidirectional (RL-BT);
- bidirectional (RL-TB);
- directional (LR);
- directional (RL);
- directional (BT);
- directional (TB);
- by colour;
- by block;
- by block (LR);
- by block (RL);
- by block (TB);
- by block (BT);
- two by two.

If a correlation is present between the correlation value and the scaled performance, this would give indication of the strategies having an influence on the operator performance. That the performance is scaled does not effect this, as per operator this only is a constant division factor equal to the mean performance. Trend lines are indicated by fitting a line using least squares optimisation. Note that due to some strategies having a very constant value, not all fits are suitable as an indication of trends. While many trend lines do indicate positive or negative correlation of strategy correlation to assembly performance, manual inspection often results in either the fit being very poor or the number of samples being small. Some strategies also seem to have clusters on specific and limited values of the correlation value, decreasing the usefulness of scattering the results like depicted.

Positive and negative correlations are both of interest, as positive correlation indicates that following the strategy is bad for assembly time, and negative correlation indicates that following the strategy is beneficial for assembly time. Horizontal trend lines would therefore indicate no correlation present between the strategy and performance. Significant variability is expected, as Figure 7.9 has shown that used strategies for the (relative) fastest and slowest assemblies of different products are not consistent.

An overview of the correlations is given in Table 7.3 with manual adaptions, to filter out incorrect fittings of the trend lines. For example, a single sample value can influence the fit drastically. Also, variance around the same correlation value and lacking data of other correlation values, influences the fit drastically, refer for example to bottom to top assembly and top to bottom assembly by block in Figure 7.14, for operator six. The process of checking the influence could be automated by checking the influence of single samples, and if this is to significant, not taking these samples into account. Overall, the variation in the performance levels is so big, that it is not possible to manually assess if the found trend lines are caused by actual correlation of the strategy to performance, or caused by variation. Possibly the variation even overshadows the existing correlations. Looking at the results, the previous results do not line up with the manual check of all separate trend lines of the single strategy versus operator results.

Table 7.3: Overview of manual analysis of positive correlation marked with a + and negative correlation marked with a - between strategy correlation and scaled assembly time, for every operator and every strategy

		operator	1	2	3	4	5	6	7	8	9	10	11	12
strategy	**marker**	**figure**												
bidirectional														
LR-BT	●	Figure 7.14				+	-							
LR-TB	●	Figure 7.15												
RL-BT	●	Figure 7.16				+								
RL-TB	●	Figure 7.14				+	+	+						
directional														
LR	●	Figure 7.15												
RL	●	Figure 7.16	+			+		+						
BT	●	Figure 7.14	-		+				-			-		
TB	●	Figure 7.15												
by colour	●	Figure 7.16												
by block	●	Figure 7.14												
LR	●	Figure 7.15												
RL	●	Figure 7.16	+											
TB	●	Figure 7.14					-							
BT	●	Figure 7.15	-											
two by two	●	Figure 7.16					-							

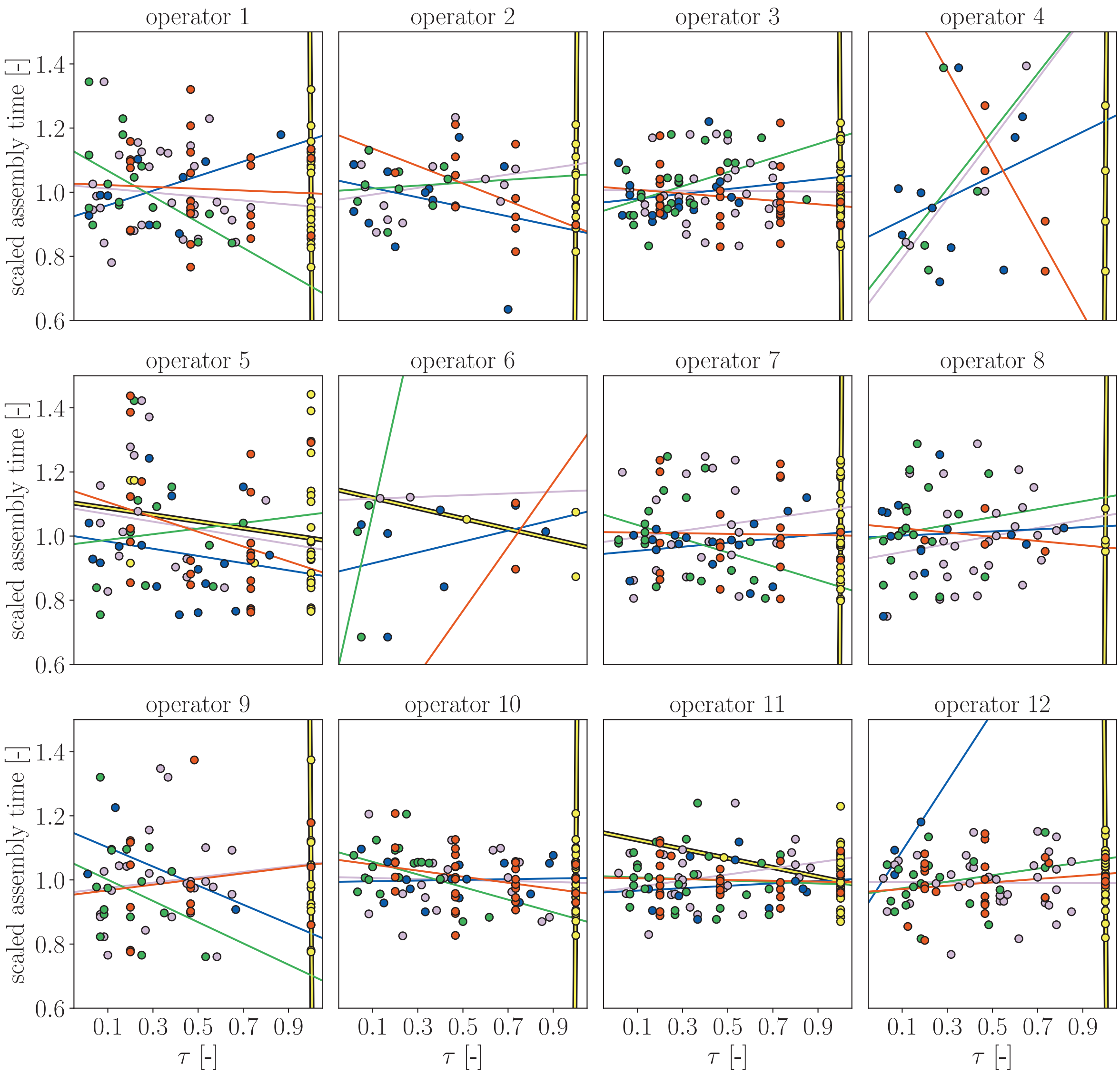


Figure 7.14: Assembly performance depicted against strategy correlation τ, part I

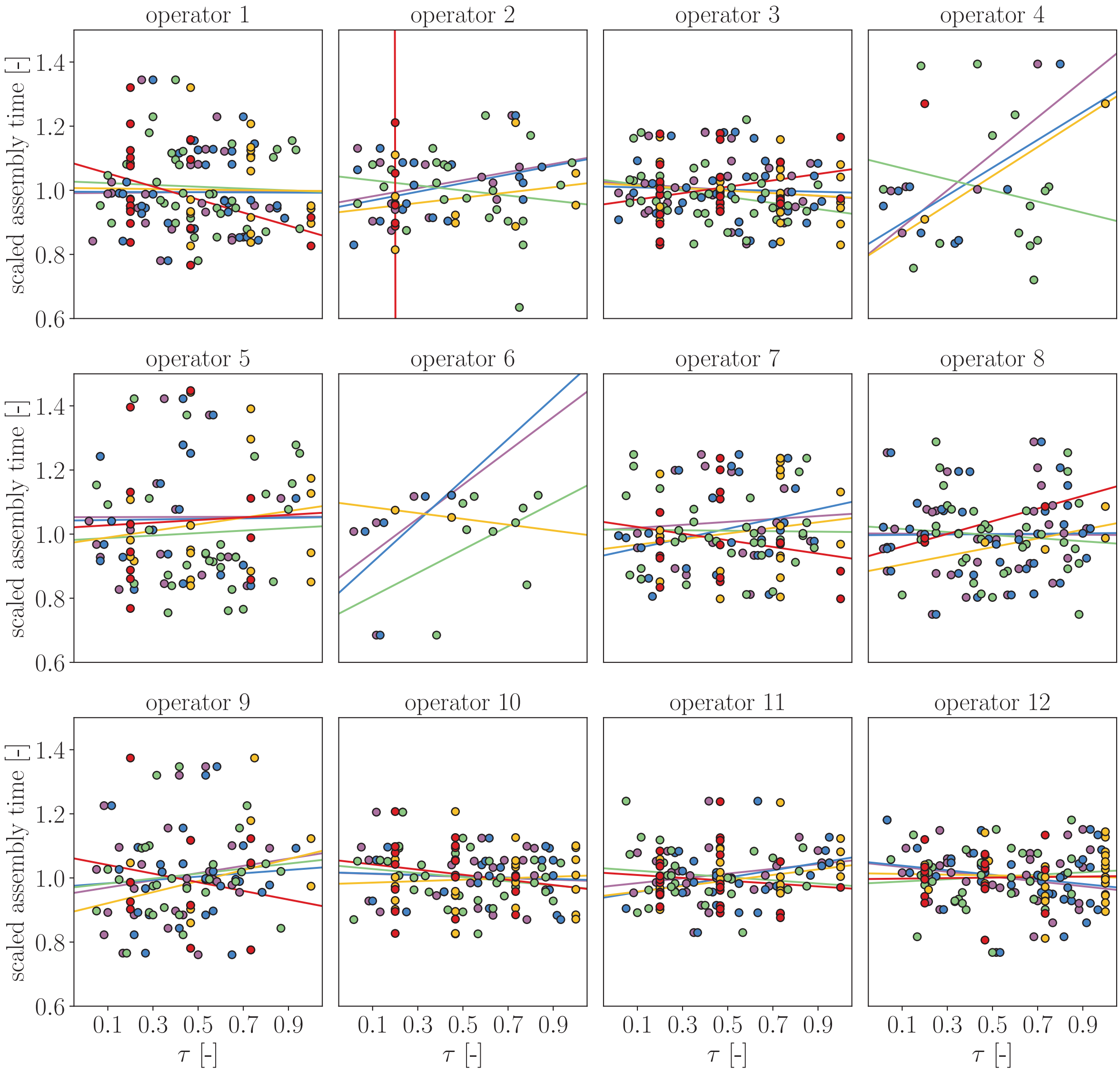


Figure 7.15: Assembly performance depicted against strategy correlation τ, part II

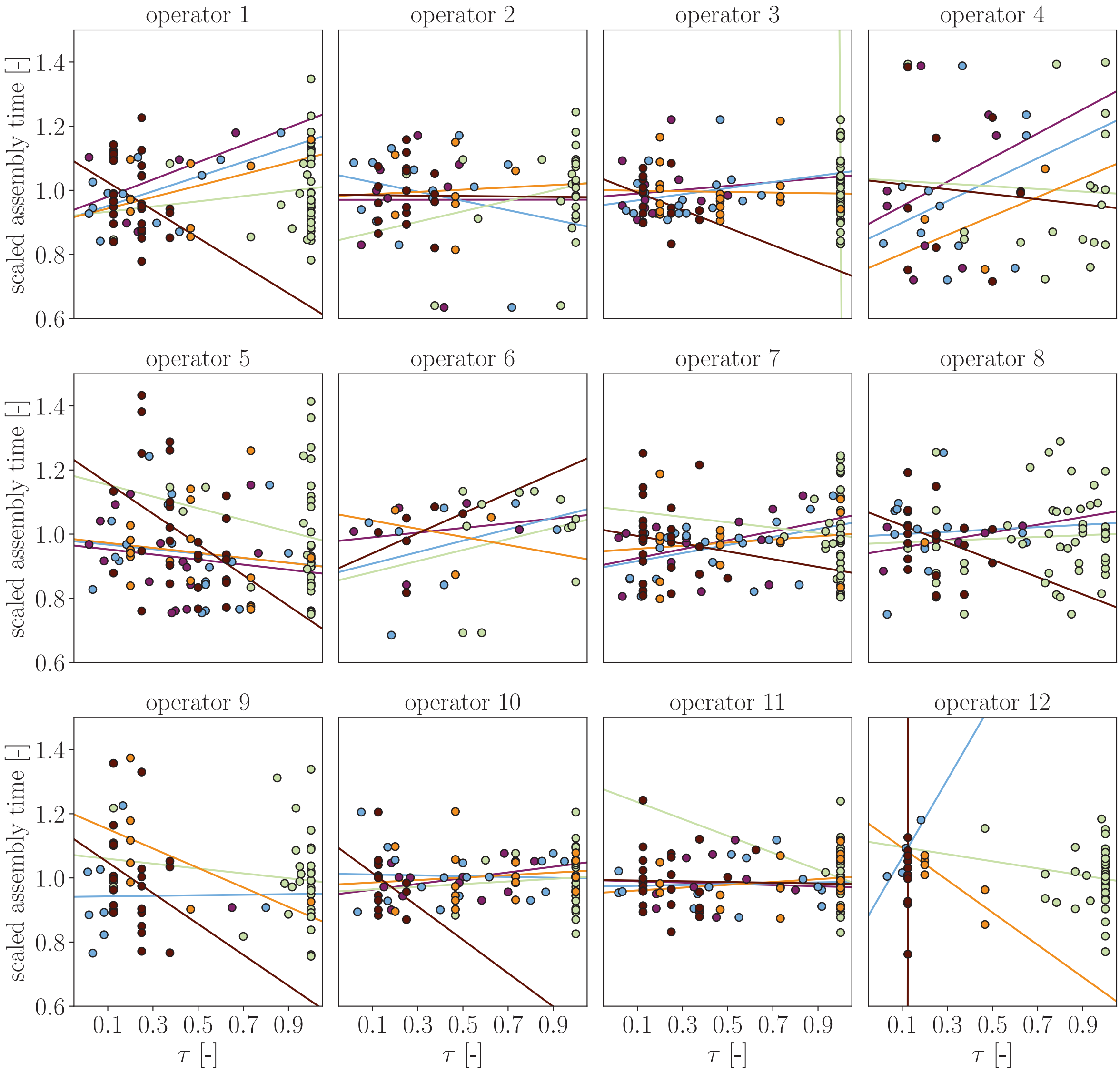


Figure 7.16: Assembly performance depicted against strategy correlation τ, part III

7.6 Conclusion

Global assembly strategies are defined, including a correlation measure to these strategies based on the assembly order of the products. Overall, it can be concluded that operators mostly assemble colour by colour and by block, but the correlation to these strategies and others is not consistent between products as it has been shown that the correlations to strategies are unequal for different products. Correlation of strategy correlation and assembly performance could not be shown and a comparison between the extreme cases of assembly, the relative fastest and slowest realisations, also resulted in limited differences in strategy correlation realisations. Possibly, increasing the difficulty of the products would increase the usage and necessity of using strategies, which could lead to more correlated results. Overall, the approach presented does certainly have potential as it has proven to perform as expected; indicating strategy following and correlating the strategy usage to assembly performance.

Chapter 8 introduces strategies based upon placement orders of bolts within blocks of block based products. As these are more often repeated, sample sizes increase which increases the potential of being able to statistically show differences in assembly strategies used. Correlating the local strategies to performance is explored too.

Chapter 8

Local Strategies

Global strategies describe the assembly of a complete product. In this chapter, *local* strategies are introduced, which make use of the introduced blocks in section 4.5, and describe the assembly strategies of the blocks. All blocks consist out of four bolts, which means only a concise set of 24 bolt orders is required to describe all block assemblies. This allows an analysis of all assembled blocks and their assembly orders including an analysis concerning operators' different assembly strategy preferences, which are potentially dependent on the block features introduced in section 4.5. If differences are found, they can potentially be correlated to performance. Then the high performing strategies could be translated to assembly instructions to improve assembly performance.

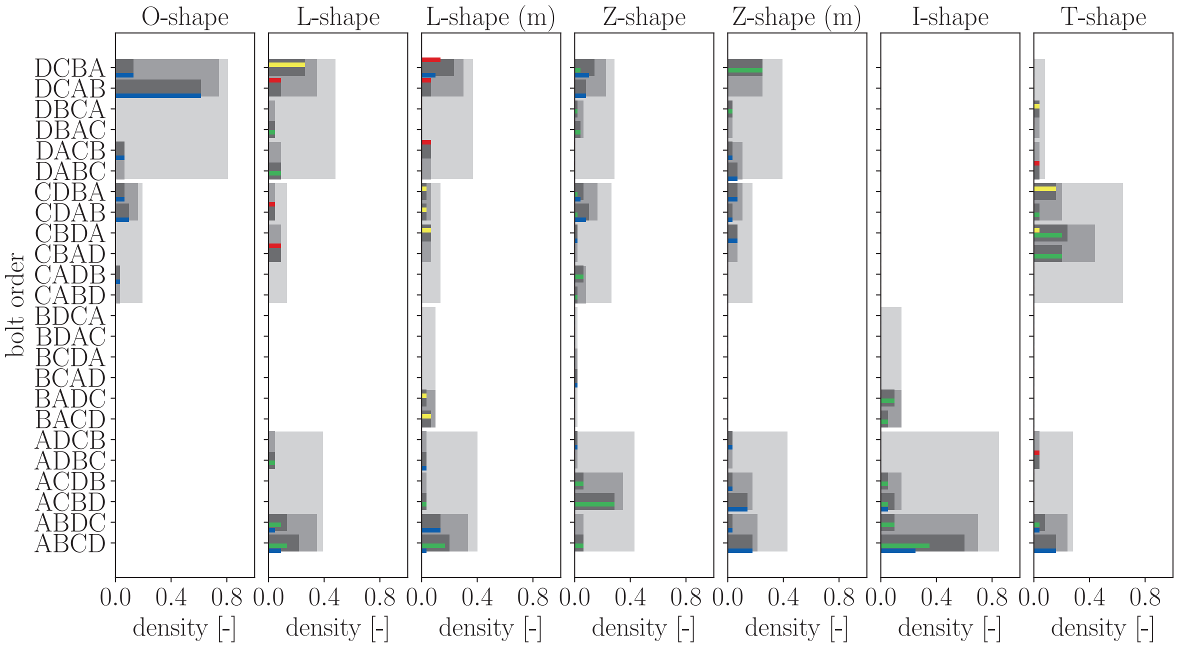
8.1 Local Strategy Visualisations Based Upon Features

As has been introduced in section 6.6.2, operator preferences based on the bolt placement order have been found. As the blocks are repeated often in the products, a high sample size of the block shapes is available, and to an extend of a second feature too, to further specify a comparison to be made for analysis. If it can be shown that at least some operators structurally choose for a certain local block based strategy on bolt level with performance differences between strategies, strategies can be linked to performance to find optimal strategies that can be applied to other operators if they are not already using that strategy. Of course, it remains to be seen then if that strategy is also more efficient for that operator. Separating the preference of strategies and the effect of strategies on performance also allows to analyse if operators actually prefer the fastest assembly strategies by themselves, to show the necessity of letting operators try a different local strategy which is expected to be more efficient, even when they do not already assemble the blocks like that by themselves. Therefore, first the operator preferences for local strategies are explored.

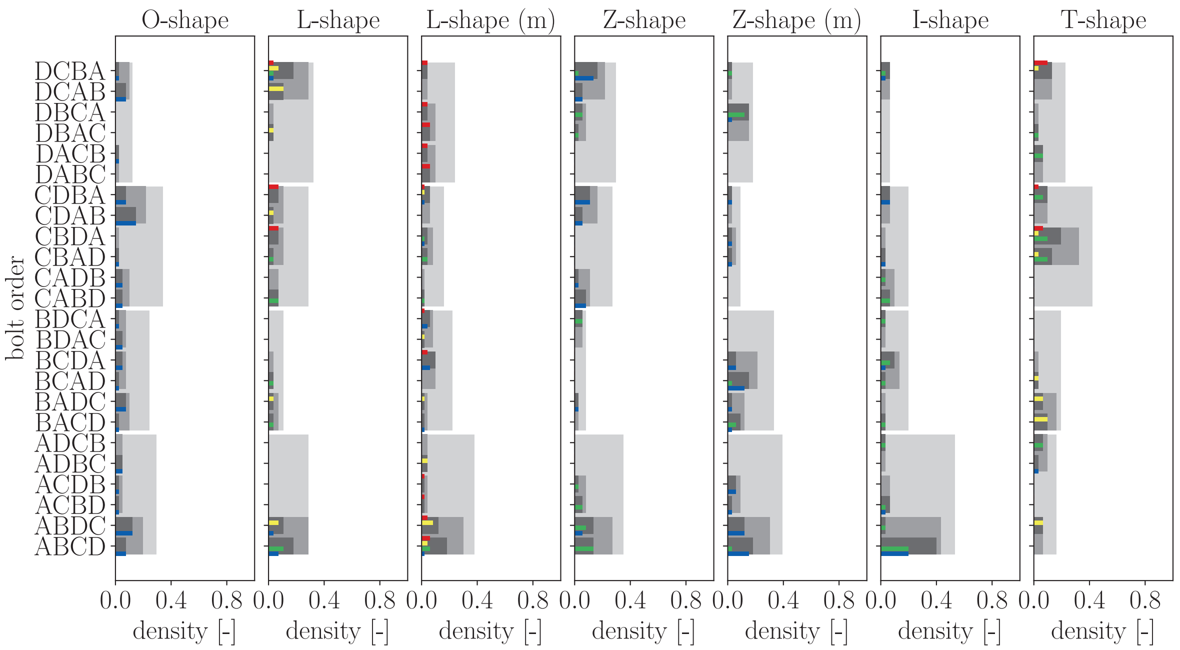
For visualisation readability, the statistics are sorted by block and by operator. Figure 8.1a shows the statistics of operator three and Figure 8.1b shows the statistics of operator nine, both with further specification based on block orientation. Bars depict the fraction of the assemblies that have been performed in that order. Bars in the colour ■ have the same starting bolt, bars in the colour ■ have the same two starting bolts and bars in the colour ■ are unique orders. Hues are used to indicate a third variable, in this case block orientations, within the unique orders. Note that the total of a bar type within another bar type sums up to the latter bars total. If non-unique placement orders are detected, all unique bolt placement orders that combined can form this non-unique one, are given an additional occurrence. A comparison of this approach versus disregarding occurrences of non-unique assembly orders is given in section 8.2.

For the block orientation based figures, colour ■ represents 0°, colour ■ 90°, colour ■ 180° and colour ■ 270°. Some interesting observations are the differences in variability of the different strategies used. Based on previous results, preferences for assembly orders are expected as section 6.6.2 has shown unique ordering could be recognised within blocks further specified to single feature values. Operator three has some clear preferences, certainly for the O-shape and T-shape, while operator nine does not seem to have preferences. Concerning the influence of block orientation, a relation between the orientation and the chosen placement order is expected if different bars of colour ■ have different dominant orientation densities. For operator three, Z-shape mirrored it can for example be clearly seen that most blocks with orientation 90° are assembled beginning with bolts *DC* or *DB*, while other orientations mostly start with bolts *CD*, *CB*, or start with bolt *A*. Note that unique assembly orders with a low absolute density value, have limited occurrences. Summarising, it is clear that at least some operators assemble using different strategies, which indicates potential of directing operators to the most efficient strategy if certain strategies can be linked to better performance, possibly dependent on block features.

Switching from block orientation to block x-coordinates gives the results as depicted in Figure 8.2a and Figure 8.2b for respectively operator three and nine. Colour ■ represents $x = 1$, colour ■ $x = 2$, colour ■ $x = 3$, colour ■ $x = 4$, colour ■ $x = 5$, colour ■ $x = 6$ and colour ■ $x = 7$. Here, the grey bars are of course equal as they represent the same data as their block orientation based counterparts depicted in respectively Figure 8.1a and Figure 8.1b. If the block x-coordinates influence preferred assembly order, substantial occurring assembly orders should have a clear (relatively) high density of a specific coordinate as this would indicate assembly preference based on x-coordinate. For both operators, abundant occurrences of specific x-coordinates within substantial high density bolt orders are not present, indicating the x-coordinate of the blocks does not influence the assembly. Analysis of the other features did not yield different results concerning the feature values having an effect on the local strategy preferences, and are therefore not depicted.

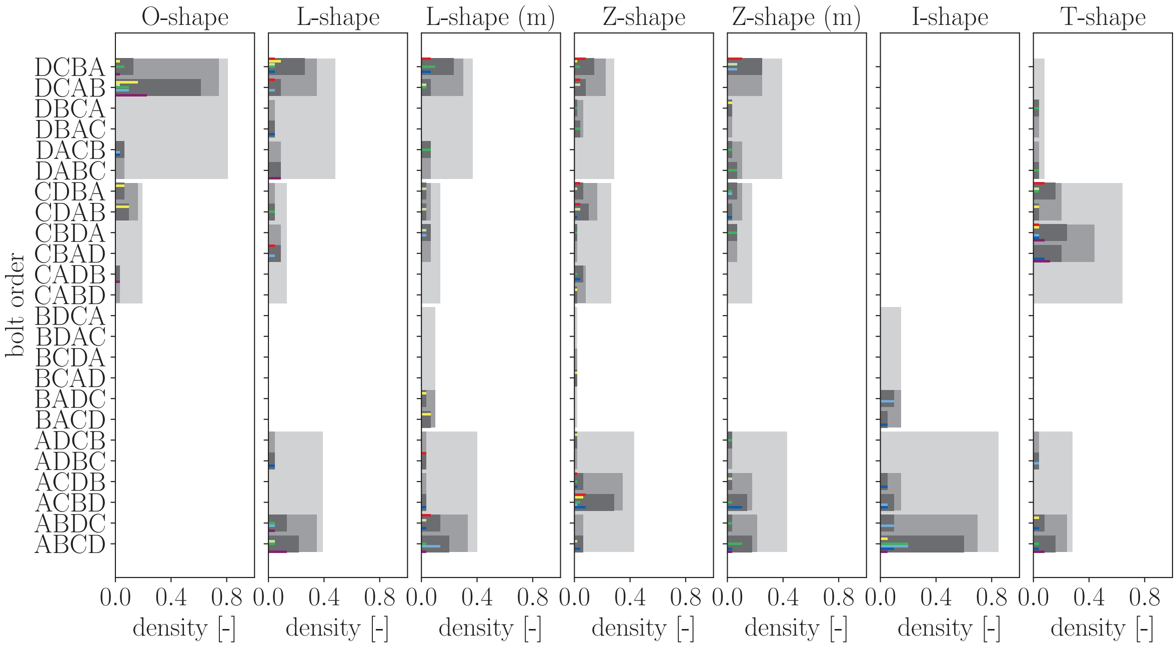


(a) Operator three

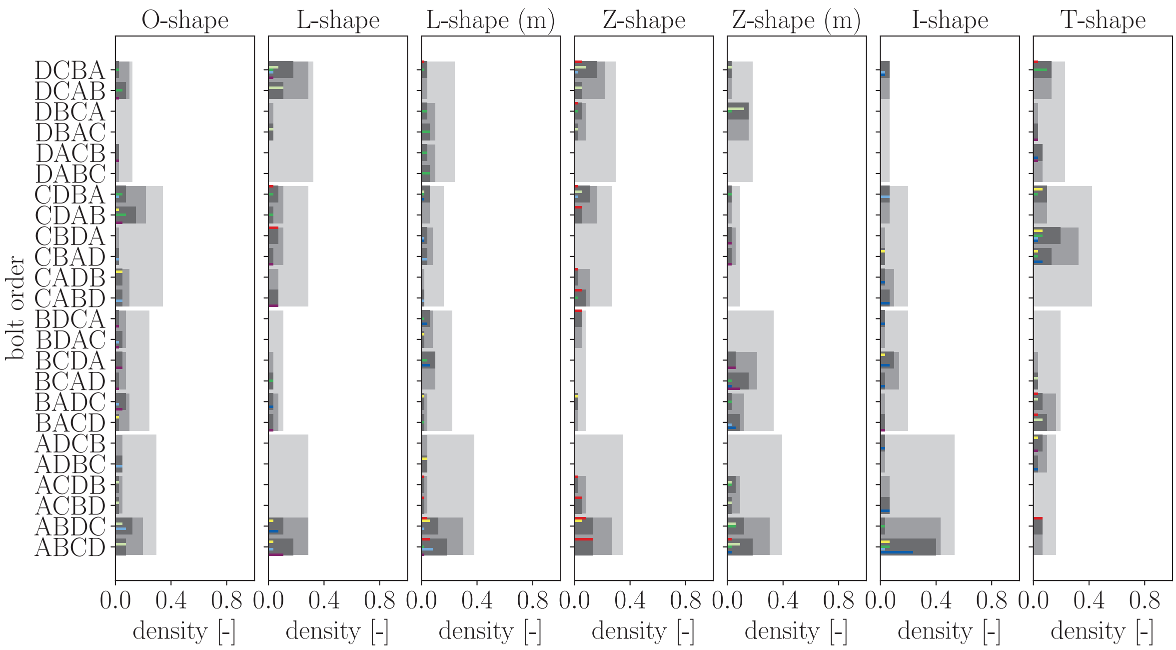


(b) Operator nine

Figure 8.1: Local strategy statistics sorted by bolt placement order, hues represent block orientation in increasing order from bottom to top with the grey bars summarising the density of all included bolt orders. Non-unique placement orders are added to all possible unique representations of orders



(a) Operator three



(b) Operator nine

Figure 8.2: Local strategy statistics sorted by bolt placement order, hues represent blocks' x-coordinates in increasing order from bottom to top with the grey bars summarising the density of all included bolt orders. Non-unique placement orders are added to all possible unique representations of orders

A variant of the visualisations based on local strategies concerning bolt placement orders can be made by not sorting in starting order, but in ending order. An ending preference visualisation based on the same data as in Figure 8.1a is depicted in Figure 8.3. Now, the way operators like to end their assemblies can much better be analysed. Again preferences are expected, since operators have been observed in section 6.6.2 to assemble in specific orders. Figure 8.1a showed that operator three likes to begin with bolts D and C of an O-shape. Figure 8.3 additionally gives the information that operator three also likes to assemble bolt B last and bolt A second to last. Figure 8.1a also holds this information, but some more effort has to be made to arrive at the same conclusion by comparing all bars and observing that all high occurrences also end with the same bolts.

While the comparison of operator three and nine is one of the more contrasting of all combinations of operators possible, this comparison shows a clear sign of operator differences that for example could be used to recognise operators based on assembly patterns of common parts in products. As introduced, the same analysis of the data as in this section, but now based on unique occurrences of assembly orders, is performed next.

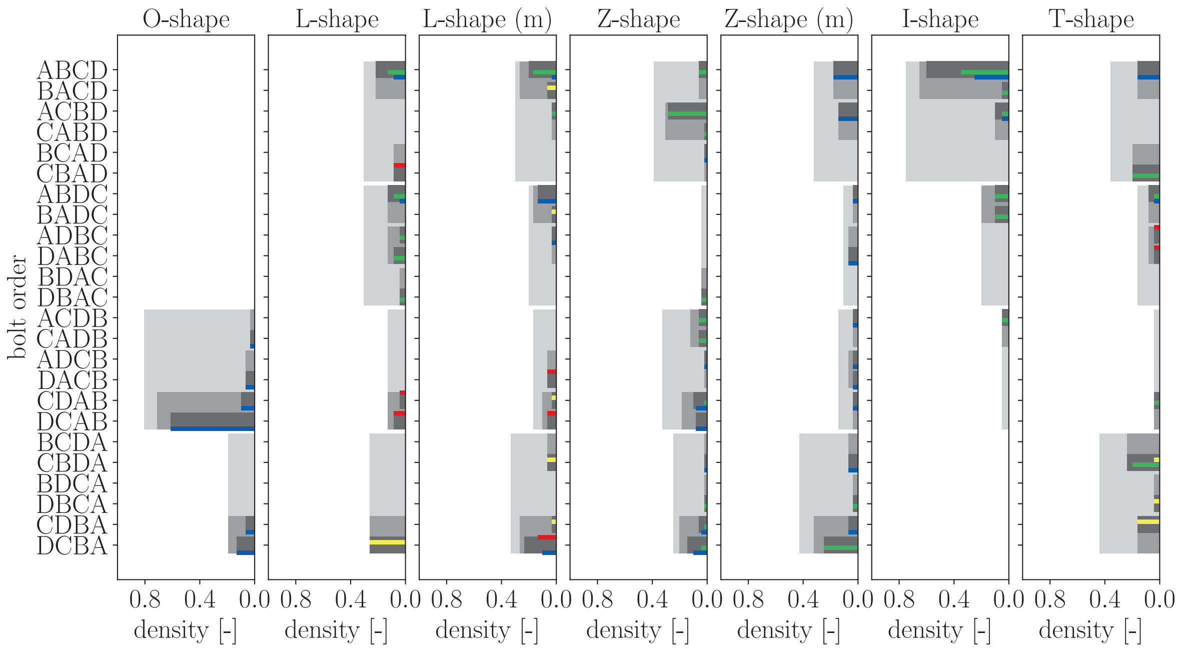
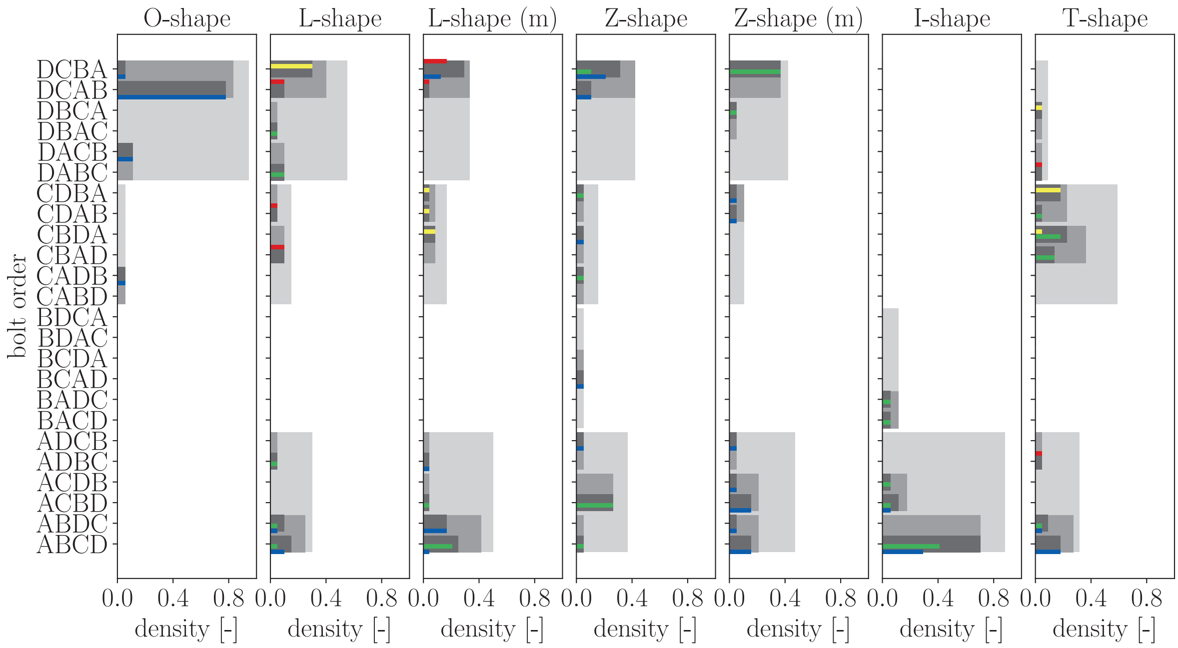


Figure 8.3: Local strategy statistics operator three sorted by reversed bolt placement order, hues represent block orientation in increasing order from bottom to top with the grey bars summarising the density of all included bolt orders. Non-unique placement orders are added to all possible unique representations of orders

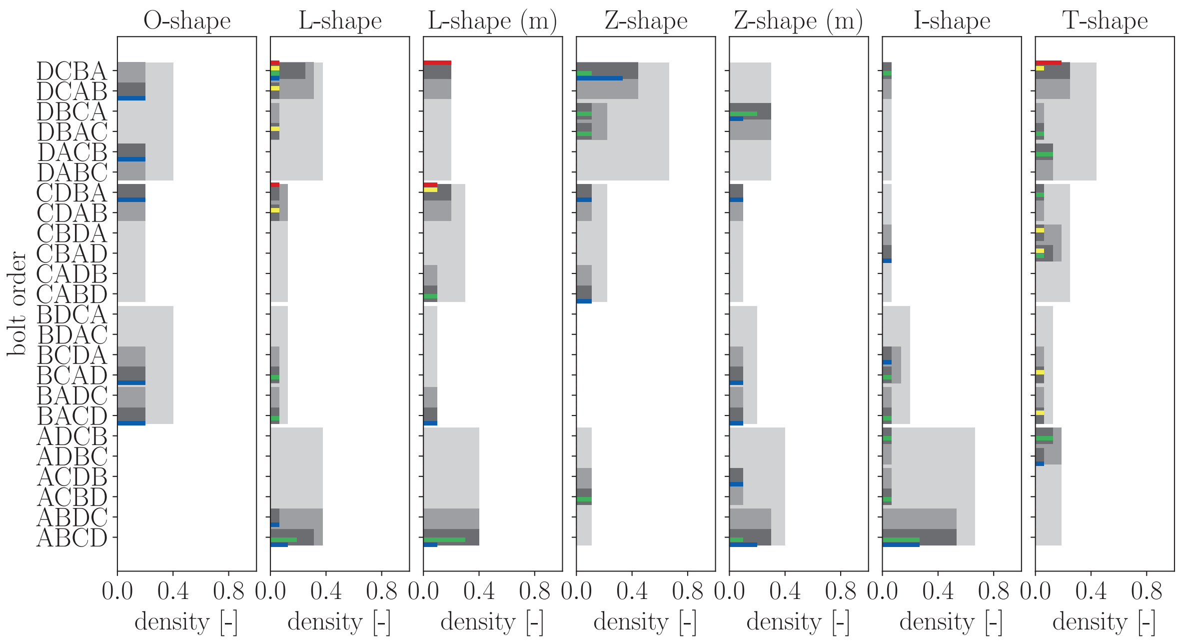
8.2 Unique Occurrences of Strategies

Figure 8.4a shows the same data as Figure 8.1a: a comparison of operator three based on block orientation, with the exception that Figure 8.4a does not count occurrences of local strategies that are not unique, to only show unique orders that have actually been assembled by operators. Parallel placing operators will therefore decrease in sample occurrences, as parallel placed bolts define non-unique bolt orders. Also, no obervations are counted multiple times anymore. Differences are minor, but the preferences of assembling an O-shape in the order $DCAB$ by operator three is even more clear now, potentially indicating that the detected non-uniqueness was caused by occlusions if operator three did actually pretty much always assemble in that order. Unfortunately, the non-uniqueness caused by parallel placement of bolts could however also have occurred more often in reality, but just not detected due the same possible occlusions. An indication of occlusion of the assembly platform as additional data could help conclude which of these explanations is the truth. Figure 8.4b depicts the same figure type, but for operator nine. Still less preference for strategies is observed than for operator three, but some strategies do stand out way more now as preferred. It is therefore clear that the uncertainty concerning the correctness of assembly orders should be fixed, before final conclusions can be made concerning these strategies, as the two presented methods of including and excluding non-unique orders differ enough to make a difference in these conclusions.

The next section presents the scaled operator performance depicted against the local strategies, to find out if local strategies can be applied in optimising assembly instructions within blocks.



(a) Operator three



(b) Operator nine

Figure 8.4: Local strategy statistics sorted by bolt placement order, hues represent block orientation in increasing order from bottom to top with the grey bars summarising the density of all included bolt orders. Non-unique placement orders are disregarded

8.3 Local Strategy Correlation to Performance

Having introduced the operator strategies, and therefore preferences of assembly, the next step is to analyse the performance correlation to these strategies per operator. Again operator nine is taken, with Figure 8.5 depicting the results with non-unique placement orders depicted for all unique representations of the non-unique order, and Figure 8.6 disregarding data with non-unique placement orders. The abbreviation *ass. time* stands for assembly time. The hues represent the block orientations with equal hues as introduced in section 8.1. Boxplots in colour ■ represent the data of orders with an equal first placed bolt, and boxplots in colour ■ represent the data of orders with an equal first and second placed bolt. Boxplots of the data per local strategy based on the placement order of all bolts are not depicted to not clutter the visualisation. Also, due to the low occurrences of the local strategies, boxplots do not offer additional information for most local strategies on their own. Dashed lines in colour ■ depict the median value of the corresponding boxplot they belong to, lines in colour ■ depict the mean value.

It is expected that operators' preferred assembly strategy is also the fastest, as operators are expected to prefer the fastest assembly strategy. Further factors, except boredom which has been indicated as influencing random behaviour as introduced in section 5.6, should not effect the choice of speed as preference. The downside of this is that if this is true, the other strategies have few or none occurrences, which makes statistically comparing the strategies based on assembly speed impossible.

For operator nine, data with non-unique occurrences of assembly orders is relatively spread out, as can be seen in Figure 8.1b when looking at the starting bolt index. However, statistically operator performance differs quite significantly in Figure 8.5. For example, starting with index A seemingly leads to faster assembly. Compare however this observation with Figure 8.6, from which can be concluded that data occurrences are significantly inflated by copying non-unique occurrences of assembly orders to all unique representations. This is caused by operator nine apparently assembling in parallel a lot.

Figure 8.7 depicts the same data without non-unique occurrences, but for operator three. The O-shape and I-shape have too few occurrences of different strategies for a good comparison, but for the L-shape and mirrored Z-shape, it can be concluded that the lesser occurring strategy of beginning with index A has shorter assembly times. For the Z-shape a similar conclusion can be drawn, but the preference of operator nine between starting with index A and index D is less clear. Note that between these preferences, the orientation is quite sorted between the slower and faster strategies. Performance may therefore be correlated to the orientation of the blocks, but as consequently no data is available to check this, this should be validated when instructions are implemented.

For now it can be concluded that operator three does not consistently use the fastest strategy for assembly, while he or she is aware of the more efficient strategy as he or she performed it multiple times. There is therefore indication that instructing operators to assemble in a specific order on bolt level would be beneficial, with even his own data being useful. Comparing the absolute assembly times between operator three and nine for different local strategies is not possible due to operator nine's lack of different strategy occurrences. Data of different strategies is required due to different overall operator assembly speeds, which only makes a single operator's relative differences between different strategies useful. Using data of another operator could be beneficial if a certain operator uses a specific strategy that is faster than the fastest assembly strategy of another operator who does not use the first operator's faster assembly strategy at all. Local strategies over time are not explored as the global strategies showed that the influence of the differences in the products is too high, and products have been assembled in different orders between operators.

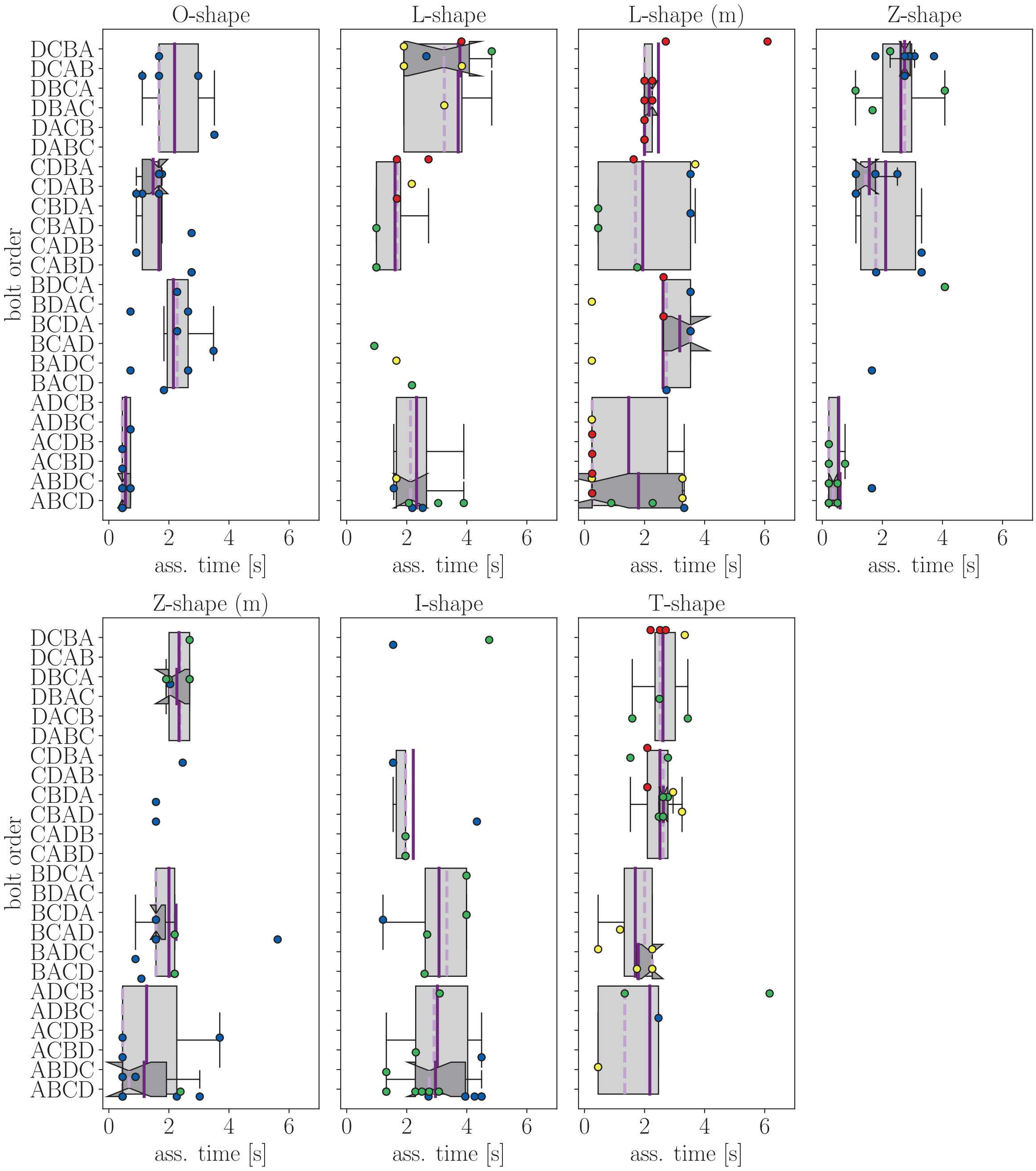


Figure 8.5: Assembly performance operator nine depicted against local strategies, hues represent block orientation in increasing order from bottom to top

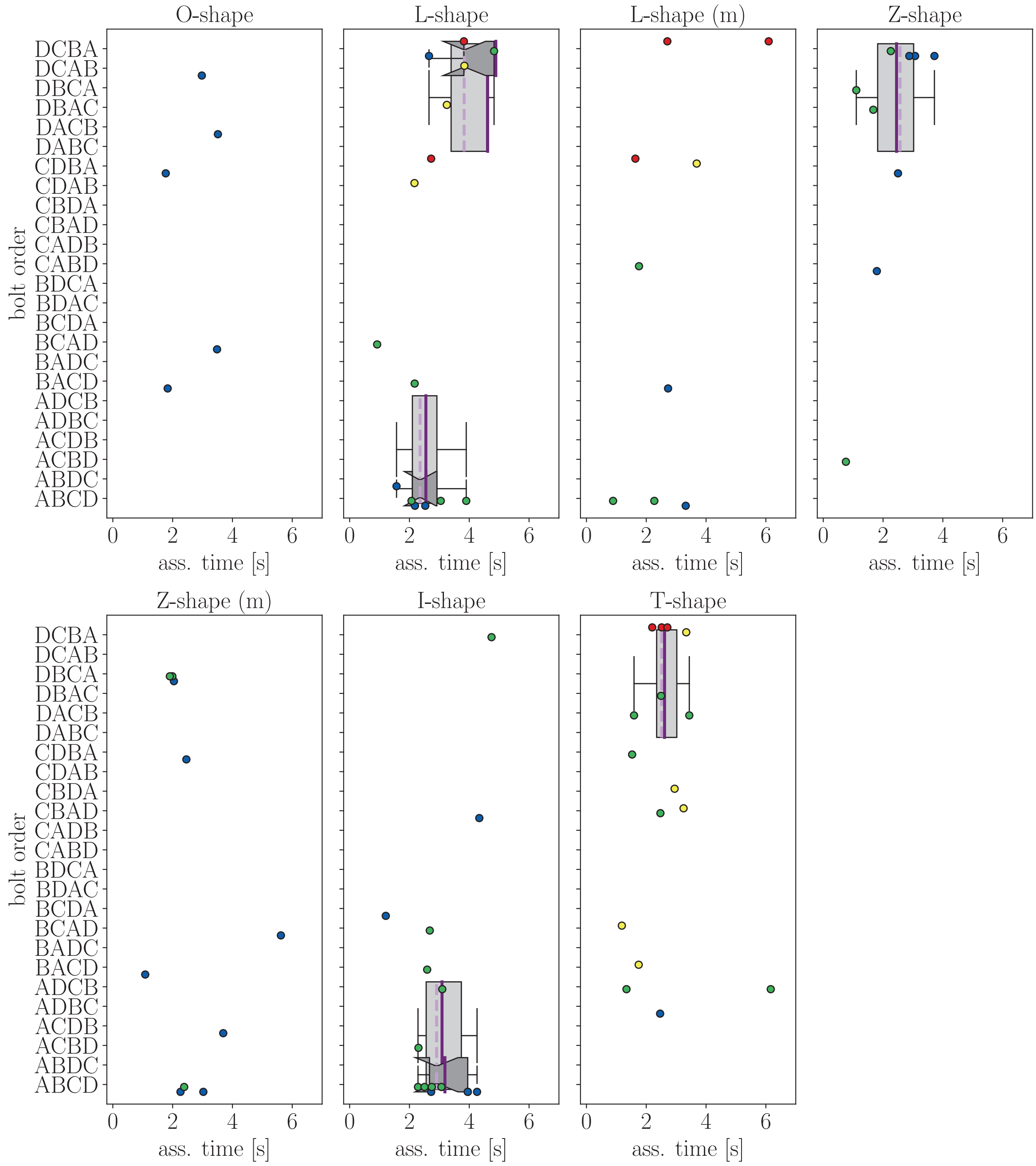


Figure 8.6: Assembly performance operator nine depicted against local strategies, hues represent block orientation in increasing order from bottom to top. Non-unique placement orders are disregarded

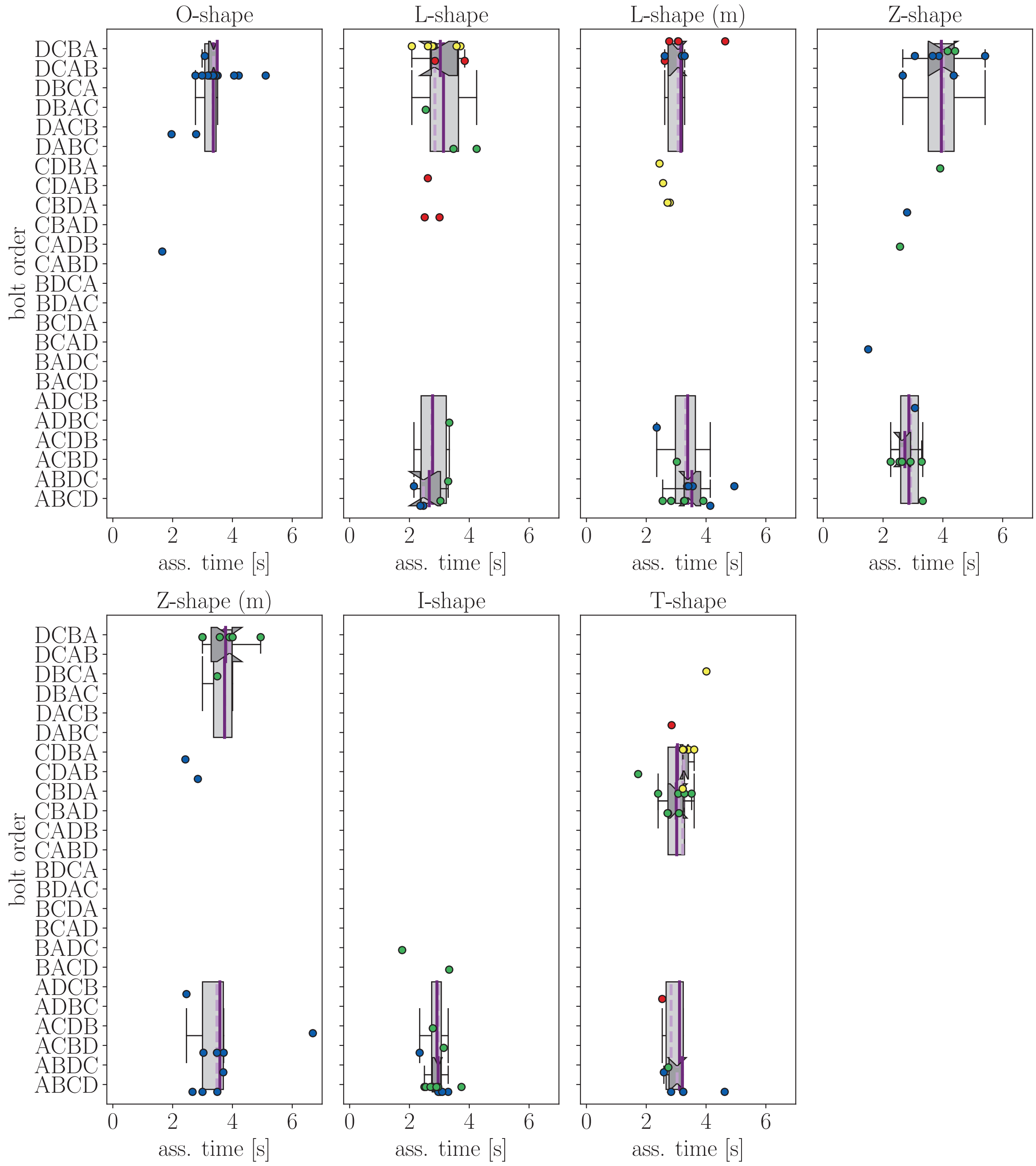


Figure 8.7: Assembly performance operator three depicted against local strategies, hues represent block orientation in increasing order from bottom to top. Non-unique placement orders are disregarded

8.4 Conclusion

Overall, it can be concluded that differences in assembly placement orders of operators can be found using a bolt level local strategy analysis. At least a sub-set of operators can be distinguished by preference for certain strategies for block shapes, the spread of the used strategies and the differences between non-unique placement results and unique placement results, based on respectively including and excluding data of parallel placement of bolts. The split of the strategies into a comparison of different depths of placement order, based on solely the first placed bolt, both the first and second bolt, and all bolts gives insights in starting preferences of operators. A variant of this analysis based on sorting the assembly placement orders from the last to first bolt, by again splitting the placement orders but now based on solely the last placed bolt, both the last and second to last bolt and all bolts, gives valuable insights in ending preferences of operators. While containing the exact same data as the previous analysis, ending preferences cannot be easily derived from the previous analysis.

Operators were expected to prefer strategies which aligned with their fastest assembly speeds. However, depicting the performance against all local strategies with again a differentiation between different placement order levels based on the placement of the first bolt, both the first and second bolt and all bolts, showed, while this effect is observable, that operators also spend a significant number of assemblies assembling by slower assembly strategies. This indicates that there is potential to give operators instructions for more efficient block placement orders on bolt level, or at least feedback concerning their performance level for a performed assembly. Due to sample size limitations of different strategies, a comparison between operator performance levels for different operators has not been performed as relative differences between strategies are of interest here due to overall operator performance differences. If assemblies with instructions are performed, all strategies could be checked to make final conclusions concerning the effect of local strategies on performance.

Chapter 9

Conclusions and Recommendations

9.1 Conclusions

In this thesis, performance differences between the assembly of high-mix and low-volume products assembled by different operators on a flexible manufacturing demonstration setup are explored. Having identified that operators assemble these products using different strategies and on different performance levels, potential has been indicated for the implementation of a learning algorithm based on gathered assembly data that adapts operators' assembly instructions to optimise assembly performance, by minimising assembly time.

Such a learning model normally makes use of previously gathered data of the process of interest, however in a high-mix and low-volume production environment, data of the to be assembled product is not guaranteed to be available, and never available in significant sample occurrence sizes. To overcome this, possible similarity measures of products are introduced which have the potential of giving a measure of similarity between two products. By identifying similar products that are already assembled to the to be assembled product, the data of these similar products can potentially be used by the learning model. The reason for this is that same optimal assembly strategy would possibly be applicable to the to be assembled product too.

The fact that the setup is a demonstration setup, allowed for a suitable measure of similarity to be implemented in the form of a set of component blocks the products are constructed with. These component blocks, referred to as block shapes, guarantee repetition in the products while still allowing variability in the complete product. Assembly is performed by physically assembling the bolt based products on a platform with a grid pattern of positions where the bolts can be placed in. Performance is measured by storing the placement time, position and colour of the placed bolts in a database, including the operator who assembled the product. A statistical analysis of assembly performance is performed, and because of the repeating blocks' features based on the shape, colour, orientation and position of the blocks, enough sample occurrences of the features are available to draw conclusions. The blocks are based on Tetris® blocks as these can all be represented by the same number of bolts, and are all uniquely identifiable. This allows for a block feature based analysis of the full product, and a local analysis of the assembly of the blocks itself.

For the purpose of the data analysis, twelve operators each assembled over 40 block based products, and a subset of the operators additionally assembled ten randomly generated products for an assembly strategy comparison between products with a structured and random layout. However, operators do have different overall assembly performance levels making it impossible to directly compare the impact of assembly strategies on absolute performance between operators. Therefore, performance can be scaled to set the mean assembly times of the performance based on the feature values being compared to an equal value. First an analysis based on a single feature was performed, which did not yield statistically different results for different operators.

Combing features did show differences in performance, but these were very limited in comparison to the number of features and operators. In addition, fixating two feature values limited the sample occurrence, making statistical conclusions difficult. There is however indication that the distance between the position of assembly and the storage bins, influences assembly performance significantly. Clear differences are found in the analysis of the assembly order of the blocks, based on the unique indices of the four bolts that make a block. An analysis of the block shapes per block orientation for all operators showed clear distinct patterns for assembly in certain orders, as placement orders of the indices are different between operators and some operators assemble bolts one by one, and others two at once.

Preferences in assembly order of blocks based on the blocks' coordinates and colours have been found, indicating directional preferences in assembly and an effect of the bolt storage colouring order. An extension of colour based influences to its effect on operator performance yielded no significant results. Operator performance has been compared to known learning curve models and yielded minimal indication of operators experiencing a learning curve, but more extensively on product than on block level, caused by the high variability in assembly time in comparison to its small mean value. Clustering of the operator block performance based on the blocks' bolt indices yielded limited correlation of features to the found clusters.

Based on the promising results concerning bolt placement orders, a strategy analysis based on product level and block level has been performed. A global strategy analysis, defining product level strategies, introduced various assembly strategies operators indicated to use, which are directional based, block based and colour based. The benefit of such an analysis is the possibility of describing all assemblies of all products in terms of these strategies. To allow analysing operators that follow imperfect strategies, a correlation measure to following the strategies is introduced, based on Kendalls τ correlation coefficient [45]. While different operators preferred different strategies, and operators differed in assembly strategies between products, limited correlation differences have been found between high and low performing assemblies. Additionally, a direct comparison between operators' correlation coefficient to strategies and the scaled performance level of that assembly, yielded inconclusive results and no significant indication of these correlations existing.

Local strategies define the exact placement order of bolts within blocks. Clear differences between operators are observed based on assembly preference of specific placement orders. It was expected that the most occurring strategies would also be the fastest, but analyses of the performance levels of all strategies indicated this was not the case. Unfortunately, the lesser used strategies often have too few sample occurrences to make correct statistical comparisons. Nevertheless, this indicates that operators' own data can potentially help improve assembly performance by indicating the faster assembly strategies using a to-be-developed learning and instruction algorithm.

Finally, using the overall conclusions, the answer to the main research question as given next can be given.

"How much potential is there for self-learning of an assembly instruction algorithm to improve the efficiency of a flexible manufacturing assembly station in a high-mix, low-volume and operator-based production environment?"

The data analysis has shown operator differences exist and it has been shown that operators use different local strategies, and to a lesser extent global strategies for assemblies. Local strategies have been shown to significantly differ in performance, but it cannot yet be concluded if feature values are correlated to this performance difference. The strategies are applicable to all variants of products in the high-mix environment, and can be applied to never before assembled products to comply to the low-volume condition. Therefore, there is potential of either applying available assembly strategy and performance data of known operators in an algorithm that generates suitable assembly instructions to direct the operator to his optimal performance strategies, link other operators' data to this operator or link a new operator with a limited data set available to a known operator and apply the most similar operator's optimal instructions to the new operator.

The answers to the sub-questions within this research are given below:

Can the conclusions of the previous research by Stellas [8] be recreated with a new, more extensive data set?
The previous results of Stellas [8] have been validated as presented in Appendix F and it has been confirmed that the summary based analysis of the product assembly performance is not suitable to be adapted to the high-mix and low-volume environment in this research.

Can a learning curve of operators be indicated?
Existing learning curve models are fitted on the assembly performance data, and exist, but barely. It is

expected that due to the simplicity of the products, the learning curve is not steep. In the case of one operator, even an increase in assembly time over time is observed. No data is disregarded based on its corresponding product or block order in the product set assembled.

Are there significant differences between the performance of different operators assembling similar products?
Assembly performance for single operators is not consistent, and there are overall differences between operators. This first observation indicates that there is potential to optimise assembly with operators' own data. The second observation indicates that are may be potential to optimise operators' performance using data of other operators. Of course, different operators can have different overall performance, but if the cause of the differences in performance lays elsewhere, potentially the performance can be increased using data of other operators. One of these potential causes is a difference in assembly usage.

Do operators have different assembly strategy preferences?
While colour by colour and block by block assembly are significantly the most popular assembly strategies, differences in strategies have been observed between operators and between products, both within blocks and within complete products. The next step is to check if this preference is caused by the preferred strategies being the most efficient.

Are assembly strategies correlated to performance?
Due to the variability in the assembly performance, expected to be caused by the relative short time of assembly and the simplicity of the products, only few relations between following a strategy more strictly on product level and assembly performance have been found, but not with significant correlation. Locally, operators differ in preference, and the different strategies statistically differ in their assembly time. Also, the preference does not always line up with the fastest strategy. This indicates there is potential of increasing efficiency of operators by instructing them to better performing assembly strategies, even using solely their own data. Due to the preferences, performance data of other strategies is however often lacking, which limits comparison options.

Can custom assembly instructions be generated for operators based upon the data of other products, to improve the operators' performance?
While this is an open question, all strategies are translatable to assembly instructions as they can be adapted to all products. By combining the, local, strategies that have high correlation to high performance, instructions that indicate optimised assembly orders for operators should be possible to be generated. Also, unused strategies can be instructed to verify their performance level.

9.2 Recommendations

A logical next step for the overarching research topic this thesis contributed to, would be to develop an algorithm that uses the performance data of the different local and global strategies, and generates an assembly order of the to be assembled product based on this products features. The introduced, or new, to be defined, features should therefore be analysed against the operator strategies and these strategies' performance levels to find out if the used strategies and their performance levels also depend on these features. This thesis only touched upon the relation between the products' features and strategy usage and performance on the local level. Combining the feature values of the different blocks in a product should be explored to find a suitable representation that can be linked to strategy usage and performance.

Concerning various topics within this thesis, separate recommendations have been made for future research concerning the improvement of assembly performance of operators in a high-mix, low-volume production environment. These are summarised by the following topics.

Hardware improvements demonstration setup
While running the experiments, it became clear that operators block the view of the camera to the assembly platform. This prevents tracking the placement of bolts correctly, and currently introduces incorrect placement times of bolts in the database, as bolts are still recognised by the neural network after the view is not occluded anymore. While these errors can be at least partially fixed by implementing software based checks, moving the camera backwards or sideways to give it an angled view on the platform,

would prevent the torso and head of the operator blocking the view. This would however give problems implementing hand tracking as a hand can block the view of another hand in this case, so using separate cameras for the bolt and hand tracking is expected to be ideal. The storage bins could also be adapted by storing the bolts either in fixed positions to prevent requiring the operators to grab bolts, or by dispensing the correct number of bolts required for assembly. Next to this, operators seemed to sometimes have uncomfortable standing positions, so a possibility of setting the working height would be beneficial for working comfort. The requirement of standing straight up to not block the camera did also not help for comfort.

Software improvements bolt recognition neural network
After an occlusion of the camera's view to the operator platform, bolt placements are recognised at incorrect times. However, if the camera cannot see the platform, this should be possible to log in the database too so that it is at least known when which bolt positions were blocked. In data processing, a time window can for example then be allocated to the placement time of a bolt instead of a single time of placement. Using this kind of logic should also then also allow to know if losing the tracking of a bolt is caused by an occlusion or an actual removal of the bolt by the operator.

Software improvements demonstration setup
It was decided that during the experiments, the operators were required to disassemble the products themselves as fastening, loosening and disassembly of the products by the collaborative robots would take too long in comparison to the time the operators would need for the assembly of a product. If both the collaborative robots can be programmed to disassemble the products, possibly the throughput of the collaborative robots is high enough that they can keep up with the assembly of the operators and also transport the bolts from the storage bins to the moving robot and back to the storage bins located at the operator. Fastening of the bolts would then still not be performed, as there is currently no added value for doing that. Furthermore, to add the automatically generated products to the demonstration setup, only the simulator mode can be used as the normal working mode of the setup cannot use side-loaded products from the database. For future experiments that require the full working capabilities of the demonstration setup, side-loading of products and assembly instructions is recommended to be added to prevent the need of manually adding these.

Implementation of hand tracking on the demonstration setup
Only tracking the placed bolts' colour, position and placement time lacks information about what happens during assembly in between placements. Picking of bolts, returning bolts, number of bolts picked, number of bolts returned and hand trajectories all give information valuable to the process of extracting used strategies by operators, but are currently unavailable for data processing. Introducing hand tracking allows to derive this information and use it to extract used assembly strategies, and therefore possibly efficiencies and inefficiencies in operator performance too.

Implementation of motion primitive identification
Using the data of hand tracking, motion primitives can be extracted from the hand trajectories. Motion primitives are a set of definitions for movement sequences which can describe the full assembly trajectory of an operator. By definition, some of these motion primitives are bad for assembly performance, and therefore operators can be compared on efficiency of their assembly process, which could give precise indications in which parts of the assembly process the operator could improve, or which part of an operator's assembly process can be used to improve other operators.

Definition of a more general similarity measure
While the block based similarity measure is suitable for the products defined in this thesis, as introduced in section 4.5, this way of defining similarity in products for statistical purposes is specific to this product. Graph based similarity as introduced in section 4.4 has more uniformly adaptable applications, as general components of products can be represented by different nodes, and their interconnection and relative positioning can be represented by, possibly weighted, edges.

Higher variability in operator backgrounds
The current dataset is mostly based on operators with similar backgrounds, namely university students with a technical background. More variability in operators' backgrounds could be beneficial for a more general comparison. Also, all operators were Dutch, who have a clear right to left directional preference

in their culture, which therefore biased the results, as an analysis where directionality is involved, is performed.

Increase difficulty of assembly
High variability is observed in the assemblies. While this could have various reasons, one of the expected reasons is the simplicity of the products which both causes the products to be assembled fast, and the operators assembling with limited concentration. Increasing the difficulty by for example turning of the pick-to-light instruction system and only using the human-machine interface would require the operator to have more concentration and would probably increase the assembly time, which would decrease the negative effect of variance if it stays the same order as it is currently.

Assembly in same order for all operators
In the analysis of the data, results were desired to be depicted against time to analyse operator performance over time, as mentioned in Chapter 7 and Chapter 8. However, the products were assembled in different orders or even unique to the operator. Since it was shown that assembly strategies differ per product and that taking an average over time of the strategies does not represent the assembly strategy of single products, this strategy analysis over time could not be performed.

Bibliography

[1] History.com. Ford's assembly line starts rolling. Retrieved from *https://www.history.com/this-day-in-history/fords-assembly-line-starts-rolling* , accessed October 9 2021.

[2] Innovatieprogramma 'Fabriek van de Toekomst'. Fieldlab Flexible Manufacturing [website]. Retrieved from *https://www.flexiblemanufacturing.nl/*, accessed December 8 2020.

[3] KMWE. The Flexible Manufacturing Field Lab Opened Its Doors [press release]. Retrieved from *https://www.kmwe.com/news/press-release-fieldlab-flexible-manufacturing-demonstrates-the-application-of-robotics-in-the-manufacturing-industry/*, accessed October 24 2020.

[4] G. van Rhijn and T. Bosch. *Operator-Oriented Product and Production Process Design for Manufacturing, Maintenance and Upgrading*, pages 133–149. Springer International Publishing, Cham, 2017.

[5] P. Geertsma. Wat is een flexible manufacturing system (fms)? `https://www.technischwerken.nl/kennisbank/techniek-kennis/wat-is-een-flexible-manufacturing-system-fms/`, 2013. [Online; accessed 26-August-2021].

[6] Omron. Omron Netherlands [Company Website]. Retrieved from *https://omron.nl/nl/home*, accessed October 24 2020.

[7] R. Cockx. Applying machine learning in a robot/operator collaborative assembly setup. Master's thesis, Eindhoven University of Technology, Dynamics and Control Group, Department of Mechanical Engineering, Eindhoven, The Netherlands, 2020. DC 2020.019.

[8] A. Stellas. Factory harmony demo - intelligent human-machine interaction system. Eindhoven University of Technology, Jheronimus Academy of Data Science, Data Science, 's-Hertogenbosch, The Netherlands, 2020. PDEng thesis - Confidential.

[9] G. James, D. Witten, T. Hastie., and R. Tibshirani. *An Introduction to Statistical Learning: with Applications in R.* Springer, 2013.

[10] M. Aehnelt and S. Bader. Information assistance for smart assembly stations. In *Proceedings of the 7th International Conference on Agents and Artificial Intelligence*, volume 2, page 143–150, 01 2015.

[11] A. Claeys, S. Hoedt, N. Soete, H. Van Landeghem, and J. Cottyn. Framework for evaluating cognitive support in mixed model assembly systems. *IFAC-PapersOnLine*, 48(3):924 – 929, 2015. 15th IFAC Symposium onInformation Control Problems in Manufacturing.

[12] T. Bosch, R. Könemann, H. de Cock, and G. van Rhijn. The effects of projected versus display instructions on productivity, quality and workload in a simulated assembly task. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '17, page 412–415, New York, NY, USA, 2017. Association for Computing Machinery.

[13] T. Bosch, G. van Rhijn, F. Krause, R. Könemann, E. S. Wilschut, and M. de Looze. Spatial augmented reality: a tool for operator guidance and training evaluated in five industrial case studies. In Fillia Makedon, editor, *PETRA '20: The 13th PErvasive Technologies Related to Assistive Environments Conference, Corfu, Greece, June 30 - July 3, 2020*, pages 40:1–40:7. ACM, 2020.

[14] E. Wilschut, R. Könemann, M. Murphy, G. van Rhijn, and T. Bosch. Evaluating learning approaches for product assembly: Using chunking of instructions, spatial augmented reality and display based work instructions. In *Proceedings of the 12th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '19, page 376–381, New York, NY, USA, 2019. Association for Computing Machinery.

[15] G. Miller. The magical number seven plus or minus two: some limits on our capacity for processing information. *Psychological review*, 63 2:81–97, 1956.

[16] M. Funk, A. Bächler, L. Bächler, T. Kosch, T. Heidenreich, and A. Schmidt. Working with augmented reality? a long-term analysis of in-situ instructions at the assembly workplace. In *Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments*, PETRA '17, page 222–229, New York, NY, USA, 2017. Association for Computing Machinery.

[17] N. Galaske and A. Reiner. Approach for the development of an adaptive worker assistance system based on an individualized profile data model. In C. Schlick and S. Trzcieliński, editors, *Advances in Ergonomics of Manufacturing: Managing the Enterprise of the Future*, pages 543–556, Cham, 2016. Springer International Publishing.

[18] W. Mayrhofer, P. Rupprecht, and S. Schlund. One-fits-all vs. tailor-made: User-centered workstations for field assembly with an application in aircraft parts manufacturing. *Procedia Manufacturing*, 39:149 – 157, 2019. 25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing August 9-14, 2019 | Chicago, Illinois (USA).

[19] S. Hart and L. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In P. Hancock and N. Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139 – 183. North-Holland, 1988.

[20] F. Zijlstra. *Efficiency in Work Behavior. A Design Approach for Modern Tools*. PhD thesis, Delft University of Technology, Deft University Press, Delft, The Netherlands., 1993. An optional note.

[21] J. Brooke. *"SUS-A quick and dirty usability scale." Usability evaluation in industry*. CRC Press, June 1996. ISBN: 9780748404605.

[22] W. Mayrhofer, P. Rupprecht, and S. Schlund. One-fits-all vs. tailor-made: User-centered workstations for field assembly with an application in aircraft parts manufacturing. *Procedia Manufacturing*, 39:149 – 157, 2019. 25th International Conference on Production Research Manufacturing Innovation: Cyber Physical Manufacturing August 9-14, 2019 | Chicago, Illinois (USA).

[23] R. Könemann, T. Bosch, and M. de Looze. Movement strategy and performance in a high-volume order picking workstation. *Work: A Journal of Prevention, Assessment and Rehabilitation*, 41(SUPPL.1):1311–1315, 2012.

[24] J. Hudok. Why using lean principles to improve high-mix low-volume facilities is not the first step to make [website post]. Retrieved from https://www.just-plan-it.com/smb-production-scheduling-blog/using-lean-manufacturing-principles-for-high-mix-low-volume-production, accessed July 20 2021.

[25] M. Murray. Production routing in manufacturing? `https://www.thebalancesmb.com/production-routing-2221386`, 2019. [Online; accessed 26-August-2021].

[26] Intel. Intel RealSense [Company Website]. Retrieved from *https://www.intelrealsense.com/depth-camera-d435/*, accessed December 4 2020.

[27] Omron. Omron FHV7 Series Smart Camera [company website]. Retrieved from *http://www.ia.omron.com/products/family/3740/*, accessed December 8 2020.

[28] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of box plots. In *The American Statistician*, number 1 in 32, pages 12–16. Taylor & Francis, Ltd. on behalf of the American Statistical Association, Feb. 1978.

[29] S. Chen and C. Chen. Motion primitive recognition on human guided robotic arm using machine learning. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, pages 955–960, 2019.

[30] D. Ferguson. Therbligs: The Keys to Simplifying Work. Retrieved from *http://gilbrethnetwork.tripod.com/therbligs.html*, accessed October 24 2020.

[31] H. Lin, C. Cheng, and W. Chen. Learning a pick-and-place robot task from human demonstration. In *2013 CACS International Automatic Control Conference (CACS)*, pages 312–317. IEEE, 2013.

[32] H. Lin, W. Chen, C. N. Huang, and H. W. Wang. Semantic recognition of human gestures based on spatial and temporal reasoning. In *2013 13th International Conference on Control, Automation and Systems (ICCAS 2013)*, pages 946–950, 2013.

[33] Q. Jiang, M. Liu, X. Wang, G. Maogen, and L. Lin. Human motion segmentation and recognition using machine vision for mechanical assembly operation. *SpringerPlus*, 5, 12 2016.

[34] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. Learning movement primitive libraries through probabilistic segmentation. *The International Journal of Robotics Research*, 36(8):879–894, 2017.

[35] S. Krishnan, A. Garg, S. Patil, C. Lea, G. Hager, P. Abbeel, and K. Goldberg. Transition state clustering: Unsupervised surgical trajectory segmentation for robot learning. *The International Journal of Robotics Research*, 36(13-14):1595–1618, 2017.

[36] A. Lemme. *Bootstrapping Movement Primitives from Complex Trajectories*. PhD thesis, Technische Fakult¨at, Universit¨at Bielefeld, 8 2014. An optional note.

[37] A. Rutledge. Gestalt principles of perception. `http://andyrutledge.com/gestalt-principles-1-figure-ground-relationship.html`, 2008. [Online; accessed 10-September-2021].

[38] G. Navarro. *A guided tour to approximate string matching*. ACM Computing Surveys, 33(1):31-88, 2001.

[39] P.E. Black. Manhattan distance. `https://www.nist.gov/dads/HTML/manhattanDistance.html`, 2019. [Online; accessed 11-September-2021].

[40] E.W. Weisstein. Voronoi diagram. `https://mathworld.wolfram.com/VoronoiDiagram.html`, 2021. [Online; accessed 12-September-2021].

[41] E.W. Weisstein. Delaunay triangulation. `https://mathworld.wolfram.com/DelaunayTriangulation.html`, 2021. [Online; accessed 12-September-2021].

[42] T. S. Caetano, J. J. McAuley, L. Cheng, Q. V. Le, and A. J. Smola. Learning graph matching. *CoRR*, abs/0806.2890, 2008.

[43] K.G.Beob and H.H. Stein. *A spreadsheet program for making a balanced Latin Square design*. Revista Colombiana de Ciencias Pecuarias, 22(4), 591-596, 2009.

[44] M.J. Anzanello and F.S. Fogliatto. *Learning curve models and applications: literature review and research directions*. International Journal of Industrial Ergonomics, 41(5):573-583, 2011.

[45] J. Magiya. Kendall rank correlation explained. `https://towardsdatascience.com/kendall-rank-correlation-explained-dee01d99c535`, 2019. [Online; accessed 24-September-2021].

[46] SciPy. scipy.stats.kendalltau. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html`, 2021. [Online; accessed 24-August-2021].

[47] J. M. Ritchie, R. C. W. Sung, H. J. Rea, T. Lim, J. R. Corney, C. Salamon, and I. K. Howley. Automated knowledge capture in 2d and 3d design environments, 2008.

Appendix A

Additional Learning Opportunities

The original research description mentions to look into learning opportunities of a flexible manufacturing assembly station, in a high-mix, low-volume production environment. For the purpose of this research, the Omron demonstration setup was designed and built. Many opportunities have been discussed concerning smart improvements to this setup. This chapter presents a summary of possible improvements with high potential that are not further explored in this thesis.

Current Operator Recognition

Badges are used to identify users for the Omron setup, so that the setup knows which data belongs to which operator. This is not necessarily anonymous. In a real factory, in terms of privacy, it could be a problem to link assembly performance data to specific operators. Would it therefore be possible to recognise operators on solely their actions while assembling? Without the need of logging in or connecting to a central operator database where operators can be identified? Using the performance data of different operators, the potential to differentiate between them can be investigated. Recognition can be based both on assembly time and strategies used.

Statistical Modelling for Smart Testing

Much data is gathered during assembly of products, including errors made by operators. Relations and patterns could be found in this data to optimise sequential testing. Sequential testing means that for this research's products, every bolt would have to be tested one by one, which would make it time efficient to test bolts with a high potential fail rate first to safe time, assuming that when the products gets sent back all errors are corrected or that low potential errors are not even checked. During the assembly, the assistance system could also give extra attention to these high potential error steps to make sure the operator does not make the error. This method can be implemented with prediction models focused on errors instead of assembly time.

Smart Bolt Fastening and Loosening

Currently the collaborative robots fasten and loosen bolts on fixed and pre-allocated positions in the product. However, as the collaborative robots have a camera, they could automatically detect where fastening needs to be performed. First the insertion depth can be checked using vision, after which the bolt is fastened using the correct torque. Based on torque feedback, damage in a bolt's or platform's thread can also be detected and indicated.

Appendix B

Therblig's Definition of Motion Primitives

Section 3.2 introduces the principle of using motion principles to describe the assembly trajectories of operators with a limited sized set of definitions for actions by which the complete assembly can be described. The full set of Therblig's motion primitives is defined in Table B.1 [47, 30].

Table B.1: Therblig motion primitives adapted from [47]

Symbol	Name	Symbol	Name	Symbol	Name
	Search		Transport Empty		Preposition
	Find		Position		Release Load
	Select		Assemble		Unavoidable Delay
	Grasp		Use		Avoidable Delay
	Hold		Disassemble		Plan
	Transport Loaded		Inspect		Rest

Appendix C

Operator Feedback

Every operator was asked for feedback concerning the assemblies as introduced in Chapter 5, after his assembly set was completed. This chapter presents a feedback list in chronological order, and gets chronologically more extensive as it got more clear which questions to ask over time. All operators are referred to as male. Note that all statements in this overview are not facts or analysis results, but the statements of operators, which could be (unintentionally) false.

Operator 1

Unavailable.

Operator 2

Did not notice repetition, even when indicated multiple times. Operator assembled the products without much thinking and considered that mindset as the likely explanation of not noticing the repetition of products.

Operator 3

Operator thinks that more than one block having the same colour influences assembly, and has the feeling that he is working colour by colour and does not notice repetition. Did not have the feeling that he made mistakes.

Operator 4

Operator does not count the number of bolts, just picks bolts. His explanation is that it is easier to just pick too many bolts and put them back if needed. If too few were picked, for example when multiple blocks were the same colour, another hand was picked and also already bolts of the next colour. Was competitive in placing and inquired after the starting moment of the assembly time. He wondered if the time started tracking when the pick-to-light system updated or when the first bolt was placed.

Operator 5

Operator feels faster when bolts of a certain colour need to be placed at the same side as their respective storage bin. Often starts topside (high y-coordinate). Considers assembly so easy, that focus on what is being done fades. Works colour by colour, but if convenient, works bin to bin. Picks one bolt per hand, and therefore also does not need to put back bolts. When asked, if the product needs to be divided into components, the colours are mentioned. However, when asked for the number of components, four is mentioned even when in reality often multiple blocks have the same colour which means less than four colours are present in a product. Repetition of the shapes is recognised when asked, but colours are leading. Now familiar with the blocks, the operator considers the Z-shape more easy than the T-shape due to easiness of placing two bolts at once. Sometimes a colour of bolt is just picked without looking what has to be assembled. Repetition is not noticed.

Operator 6

Operator mentioned that bolts are difficult to pick from the bins. Improvements could be holders for alignment, dispensers or bins with the correct number of bolts. Overhead beamer is considered inconvenient due to body overhanging the assembly platform, which blocks the beamer. Groups of four make assembly easier, and overall assembly is viewed as simple and repetitive. Assembles block by block, starts in corner and works to other side. Does not assemble same colour after each other if more than one block has the same colour. Initially picked two or three bolts, but eventually three to five as he wanted to have enough bolts. No counting of bolts, just picks based on feeling. Picks extra or places some back if needed. Shapes are recognised, but operator had pre-knowledge. Felt that he would be faster if more bolts of the same colour could be placed if for example multiple products could be assembled in parallel. Working stance inconvenient due to beamer and camera placement. Felt it was not worth looking into which assembly strategy to use as it would probably still be faster to just assemble in an any order. Repetition was only noticed in the beginning and end of the assembly set. Operator places two bolts at once. Operator leaned forward a lot during assembly, so many of his assembly realisations are incorrectly tracked.

Operator 7

Operator starts with the most prominently present colour in amounts. Rather picks too many bolts of a certain colour than too few. Stops thinking during assembly quite fast, just follows instructions. Considers assembly quite boring and without satisfaction. Begins assembly with most common colour and if multiple blocks have the same colour they are assembled after each other. Bin order not considered important. Always picks more than four bolts to be safe as the repetition of multiples of four was noticed. bolts are returned to the bins. No motivation to be fast, just did was he was asked. No desire to try other things. Product is looked at before assembly. Repetition was barely noticed. If assembly strategy has to be indicated, probably assembly from top to bottom and in arbitrary x-direction was most prominent. Every hand placed a bolt if that was physically possible.

Operator 8

If bolts on a certain row are one colour, operator worked colour per colour, otherwise not. However, if picking two colours at once results in arms crossings, picking colour per colour is again preferred. Picked two colours at once if hands would not overlap. It would be preferred if the bins would be positioned to prevent that. If a colour is abundant, first do the other colours so that the bolts left over to place are all the same colour. For the random products, it was considered way harder to see if a bolt had been forgotten to be placed down. Counting was considered, but not done as it would cost time to do. Overall, assembly felt consistent in speed, and not much thinking was required. In the beginning assembly started from the left. However, this got boring so operator also started beginning from the right. Bin order was not considered, except when hands needed to cross. Number of bolts picked was around one or two per hand, except during the last ten random products where exactly four were picked to prevent forgetting to place a bolt. Numbers on the bins were not used for the block based products. Bolts were counted after picking, and none were placed back in bins. Blocks were not recognised when asked. Before assembly, product was looked at so assembly was not performed by blindly following the instructions. Repetition was however not noticed. If the preferred way of assembly needs to be described, assembly was performed row per row from left to right. If bolts were aligned nicely, multiple were placed at once.

Operator 9

Operator uses random assembly orders, also for equal products. Actively recognised the same products. Described assembly as chaotic, constantly changing. Thinks that there probably some optimal way of assembly, but could not follow it. Assembled colour per colour. Bin order was sometimes used. Picked four bolts preferably, sometimes three by accident. If eight of the same colour were needed, full hand of bolts was picked. No exact counting of bolt numbers, just picked based on feeling. Bolts are returned if needed. Picked with one hand, placed one by one with other hand and final two bolts were placed with both hands in parallel. It was noticed that the bolts were grouped, however not that the groups were certain shapes. In response to the clarification that certain shapes were used, the answer was that with four bolts not many more shapes than the introduced seven would be possible. Picking of bolts was not convenient, too many were located together in the bins. It was necessary to grab them. It would be better to dispense as many as needed. Did try to look at the product first before beginning, but was not of much help. Often started with the top left corner, so started with the colour which was there. Only looked at the bigger picture in the random products. Looked at how many groups of bolts of the same colour were present to know how many bolts of that colour were needed (so four, eight or twelve).

Operator 10

When the operator was concentrated, he worked left to right. Without concentrating, red bolts were placed first as red is the first bin from the left. If no red bolts were present, green bolts first were placed first. If a red block is located on the right of the platform, operator works from right to left. A random product is assembled bolt cluster per cluster. Often picks three bolts, but also four. No counting of bolts, picking was based on feeling. Bolts are returned in the bins if needed. Did notice the blocks, but had pre-knowledge of them. If not concentrated, operator checked if red was in the product by checking the number of bolts to pick on the bins, thereafter checked where to place exactly. If concentrated, checked what colour was placed most left. Looked at the colours, not at the shapes. Did not look at the product itself during assembly, but assembly of random products did help realise that groupings help. Repetition was noticed. Bolts were picked hand full and placed from this hand.

Operator 11

Operator noticed the bolt multiples of four. Indicated after finishing the assembly set that focus was needed. Commented that it was possible to have a conversation on the side during assembly. Assembly was not always on maximum speed but operator was also not dawdling. Did not notice the number of bolts on the bins initially, but found that information useful when it was noticed to know which bolt colour was most abundant. Would not have noticed if the number was incorrect. Overall, assembly was considered not hard and not complex, but as indicated already, some concentration was needed. Pick-to-light system was considered helpful for preventing errors. Worked colour per colour, no specific preferred staring colour. Did however start with a certain colour if the number of bolts equalled eight. Twelve bolts were split into two time six. First two bolts were picked, then placed in other hand and this was repeated. Bolt numbers were counted. Therefore, returning bolts was sporadic. Bolts were placed one by one by picking from the other hand, and two together for the last two as now both hands were available. When eight bolts were placed, most of the times two times four of the same colour left. For these, maybe the preferred colour was the closest colour. Did not have the feeling that assembly was directed to much. Operator was motivated by working colour per colour, which was potentially stimulated by the blocks. Tried to think as little as possible about what he was doing, but did look at the product before assembly. Repetition of the products was not noticed, repetition of the shapes of bolts, the blocks, was noticed.

Operator 12

Operator noticed the bolt multiples of four. Less noticeable in the random products, where also more concentration was needed. Block based products were assembled from left to right, random products were assembled in storage bin order from left to right, picking four bolts every time. Set of block based products was much repetition of the same sort of assembly, so became boring. Final ten random products more challenging. However, when a strategy was found, it stayed repeating work. Assembly was performed colour per colour, whether the bolts were grouped or not. No specific colour preference to start with for the block based products, but for the random products started with red bolts, which are stored in the most left storage bin. Initially no preference of assembly, maybe later on working left to right was preferred. Did not act on occurrence of eight or even twelve of the same coloured bolts, except when they were spread over multiple clusters, then working from left to right could be ignored when for example some bolts were left to already start with the second block of the same colour. Picked as many bolts as possible, preferably the exact amount, but no counting. Did return bolts when necessary. Noticed the blocks in the products. Would prefer assembly based on assembly per colour. Operator noticed that bins were a bit to small for his hands. Preference would potentially change if for example only picking of one bolt at a time would be allowed. Repetition was noticed somewhat. Bolts were picked one by one from one hand that acted as a buffer. Last two in parallel as the hand that acted like a buffer was empty.

Appendix D

Technical Details

This chapter presents technical details of the Omron demonstration setup, the product, data gathering and processing of the data. Finally, an overview of the software used is given.

D.1 Data Gathered

Currently the only variables that are used from the data gathered by the setup, are bolt placement times, bolt colours, bolt indices and operation actions (removal or addition). In addition, the current product being assembled, the currently working operator and the product layouts are stored. The setup however tracks much more information. An overview of the database tables and its variables is given in Figure D.1 which gives a clearer overview of the potential sources of data for extensions of the research. For the tables with a fixed set of variable values, the values are given.

D.2 The Product

Technical details concerning the products are presented in this section. First, the bolt indices are linked to the x- and y-coordinates of the platform, as defined in Figure 2.2. Secondly, details concerning the generation of the block based products and the random products are presented. Also, the definition of a product type that can be loaded into the database is given.

D.2.1 Bolt Indices

The database uses indices to log bolt positions. These indices are linked to physical locations on the assembly platform, as defined in Table D.1.

Table D.1: Block coordinates corresponding to the bolt indices

x-coordinate	y-coordinate	index	x-coordinate	y-coordinate	index
1	4	1	4	2	17
1	3	2	4	1	18
1	2	3	5	5	19
2	5	4	5	4	20
2	4	5	5	3	21
2	3	6	5	2	22
2	2	7	5	1	23
2	1	8	6	5	24
3	5	9	6	4	25
3	4	10	6	3	26
3	3	11	6	2	27
3	2	12	6	1	28
3	1	13	7	4	29
4	5	14	7	3	30
4	4	15	7	2	31
4	3	16			

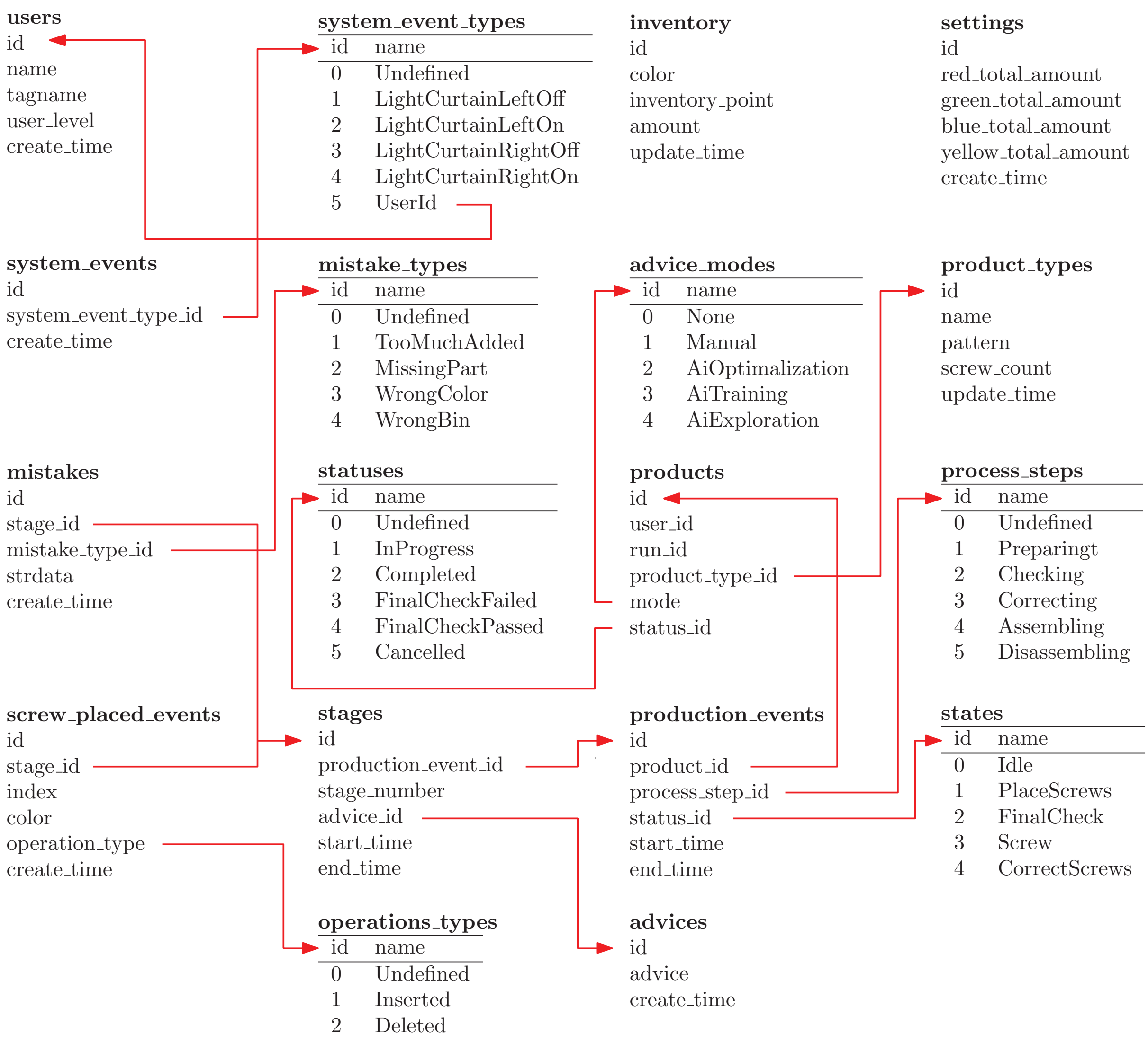


Figure D.1: Overview of the variables in the database and the linkage between them

D.2.2 Generating Products

To generate products, a randomised bolt allocation algorithm is used to place four blocks in a product. The steps taken are summarised below.

- Initialise which product types need to be generated.
- Iterate block allocation algorithm until product type is constructed consisting out of four blocks in valid positions.
- Per iteration:
  - Choose product type from random weighted distribution, to compensate for more difficult to place blocks due to layout so that the total number of all block shapes are in the same order of magnitude;
  - choose random block colour;
  - choose random block x-coordinate;
  - choose random valid block y-coordinate based on the fixed x-coordinate;
  - choose random valid block orientation;
  - check if the block's bolt coordinates do not overlap with already placed blocks' bolts and do not exceed platform size;
  - check if directly neighbouring bolts of the to be placed bolts do not overlap with already placed bolts in the same colour;
  - if all true, place block and go to next iteration until four blocks are placed.
- Add product layout to database under the product type ID.

To load the products onto the Omron setup, the resulting bolt positions and colours are put in the format as given by

$$\begin{aligned}\text{product type id}: -c_4c_9c_{14}c_{19}c_{24}-, c_1c_5c_{10}c_{15}c_{20}c_{25}c_{29}, c_2c_6c_{11}c_{16}c_{21}c_{26}c_{30},\\ c_3c_7c_{12}c_{17}c_{22}c_{27}c_{31}, -c_8c_{13}c_{18}c_{23}c_{28}-,\end{aligned} \tag{D.1}$$

with $i \in [1, 2, \ldots, 31]$ and c_i the colour of bolt index i, with R for red, G for green, B for blue and Y for yellow. As an example, the product definition of product $P_{4.9a}$, as depicted in Figure 4.9a, is given by

$$P_{4.9a}; \text{-00BB0-,YYGGB00,0YG0BRR,0YG00RR,-00000-.} \tag{D.2}$$

For the random products, 16 bolt positions are randomly chosen and randomly the colours are allocated so that each colour occurs four times.

D.3 Data Gathering

For data gathering, the simulator mode on the setup is used. The product definitions as defined in appendix D.2.2 and given by (D.1) can then be loaded in and the appropriate instruction setting can be set, which in this research case is showing the full recipe, or mode three. From the simulator, the correct product type can be set, and after starting the assembly the pick-to-light system will show the instructions. Operators have to identify themselves with the physical key pass and scan it on the reader on the setup.

To transfer the database data to a post-processing computer, a local backup can be made in *PgAdmin*, the database viewer software installed on the computer next to the setup. This backup can be loaded onto the post-processing computer using *PgAdmin* on this computer too. To access the data for post-processing, the python library *psycopg2* is used that can run *SQL queries* on the database, to load the desired data into *Python*. Installing *PgAdmin* and *Python*, including the *psycopg2* package is sufficient to load in the data.

D.4 Software

This research has been performed with the following software installs

- PgAdmin 4.28
- WinPython64-3.9.2.0
  - manual install of psycopg2 package necassary, run *pip install psycopg2* using *WinPython Command Prompt.exe*

Appendix E

Overview Feature Occurrences

Chapter 6 presents an overview of the results based on the data gathered using the experiments defined in Chapter 5. Occurrences of single feature values, as introduced in section 4.5, per operator are given by Table E.1. When multiple block features are combined, the occurrences of this specific value combination of both features can be limited and influence the visualisation as used in Chapter 6. Therefore, the occurrences of all feature value combinations possible are given in Table E.2 for operator three, Table E.3 for operator five, Table E.4 for operator six, Table E.5 for operator seven, Table E.6 for operator eight, Table E.7 for operator nine and Table E.8 for operator eleven. These are all the operators that are specifically discussed in Chapter 6. The same abbreviations of feature values are used as introduced in Chapter 6; O for O-shape, L for L-shape , L *(m)* for mirrored L-shape, Z for Z-shape, Z *(m)* for mirrored Z-shape, I for I-shape, T for T-shape, r for red, g for green, b for blue and y for yellow.

Table E.1: Overview of gathered data for every operator

operator		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
products	[-]	42	42	42	42	41	42	42	52	52	46	52	50
block		42	42	42	42	41	40	40	42	41	37	42	40
random		0	0	0	0	0	2	2	10	11	9	10	10
blocks	[-]	145	124	160	104	146	92	141	65	122	147	161	148
shape	[-]												
O-shape		21	19	21	15	18	17	22	7	13	25	22	24
L-shape		18	11	21	17	21	12	15	16	19	23	22	17
L-shape (m)		25	19	26	17	24	11	21	5	18	27	21	26
Z-shape		22	19	22	13	20	17	19	6	15	18	29	26
Z-shape (m)		22	23	29	10	20	16	15	11	17	13	21	20
I-shape		21	14	18	8	21	12	23	7	20	20	22	13
T-shape		16	19	23	24	22	7	26	13	20	21	24	22
orientation	[°]												
0		63	49	70	43	58	49	55	24	49	53	69	74
90		51	46	60	23	62	23	54	22	45	55	57	48
180		15	18	17	21	17	14	25	16	16	22	25	20
270		16	11	13	17	9	6	7	5	14	19	12	8
colour	[-]												
red		40	27	40	32	40	28	45	22	32	37	44	38
green		41	29	39	24	35	20	31	15	30	42	39	39
blue		32	39	47	23	42	30	33	14	28	35	44	39
yellow		32	29	34	25	29	14	32	16	34	35	36	34
x-coordinate	[-]												
1		22	18	19	11	23	12	18	4	17	27	15	24
2		21	17	27	18	27	8	22	8	19	20	21	25
3		19	13	25	9	22	17	21	13	16	17	30	17
4		32	26	32	28	23	11	18	14	27	25	32	27
5		16	12	16	13	17	13	26	9	16	24	23	19
6		18	19	18	15	14	12	22	15	14	24	20	22
7		17	19	23	10	20	19	14	4	15	12	22	16
y-coordinate	[-]												
1		24	25	22	22	16	9	22	11	18	19	29	25
2		34	33	41	15	36	24	37	16	33	46	42	28
3		42	27	35	30	40	34	40	17	28	46	41	43
4		36	32	41	25	42	18	28	15	30	25	28	40
5		9	7	21	12	12	7	14	8	15	13	23	14
detection success	[%]	86.3	73.8	95.2	61.9	89.0	57.5	88.1	38.7	74.4	99.3	95.8	92.5

Table E.2: Occurrences feature combinations of operator three

block		**shape**							**colour**				**orientation** [°]				**y-coordinate** [-]				
		O	L	L(m)	Z	Z(m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	3	3	6	6	6	5	11													
	g	5	6	8	2	6	7	5													
	b	6	5	9	13	5	6	3													
	y	7	7	3	8	5	0	4													
orientation [°]	0	21	3	9	12	14	6	5	18	14	19	19									
	90	0	7	6	17	8	12	10	14	17	21	8									
	180	0	6	5	0	0	0	6	5	6	2	4									
	270	0	5	6	0	0	0	2	3	2	5	3									
y-coordinate [-]	1	4	0	1	1	12	0	4	5	5	6	6	14	4	4	0					
	2	10	9	6	0	4	3	9	8	12	13	8	13	20	5	3					
	3	3	6	6	10	3	3	4	8	13	6	8	12	13	5	5					
	4	4	4	8	11	3	6	5	10	7	18	6	23	11	3	4					
	5	0	2	5	7	0	6	1	9	2	4	6	8	12	0	1					
x-coordinate [-]	1	7	4	1	0	1	1	5	4	3	5	7	10	9	0	0	0	12	3	4	0
	2	1	3	4	4	5	5	5	9	8	4	6	12	14	1	0	4	9	1	4	9
	3	4	3	5	1	3	7	2	5	3	12	5	10	11	1	3	1	5	7	6	6
	4	3	4	5	6	6	4	4	6	11	9	6	13	10	3	6	9	3	8	8	4
	5	1	2	5	4	3	0	1	2	6	6	2	6	4	5	1	5	1	6	2	2
	6	5	2	1	5	1	1	3	3	7	3	5	9	6	3	0	3	7	2	6	0
	7	0	3	5	9	3	0	3	11	1	8	3	10	6	4	3	0	4	8	11	0

Table E.3: Occurrences feature combinations of operator five

block		shape							colour				orientation [°]				y-coordinate [-]				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	5	3	7	4	4	7	10													
	g	4	3	7	5	7	5	4													
	b	6	9	5	7	5	6	4													
	y	3	6	5	4	4	3	4													
orientation [°]	0	18	8	5	8	12	4	3	17	12	21	8									
	90	0	8	7	12	8	17	10	16	15	16	15									
	180	0	5	5	0	0	0	7	3	7	2	5									
	270	0	0	7	0	0	0	2	4	1	3	1									
y-coordinate [-]	1	4	1	1	0	3	3	4	6	2	3	5	7	5	4	0					
	2	8	5	6	0	4	6	7	11	10	9	6	12	16	5	3					
	3	3	3	6	11	11	0	6	5	11	14	10	17	17	4	2					
	4	3	11	7	7	2	8	4	13	9	14	6	15	19	4	4					
	5	0	1	4	2	0	4	1	5	3	2	2	7	5	0	0					
x-coordinate [-]	1	5	10	2	0	2	0	4	6	5	8	4	12	11	0	0	0	11	1	11	0
	2	2	5	3	1	3	8	5	6	2	9	10	10	15	2	0	5	8	5	3	6
	3	4	2	4	1	3	8	0	7	6	7	2	10	10	2	0	3	4	7	7	1
	4	1	1	7	4	3	4	3	6	7	8	2	11	3	2	7	4	4	5	8	2
	5	2	1	5	3	3	0	3	4	5	4	4	5	7	4	1	2	3	11	1	0
	6	4	2	1	2	0	1	4	4	3	4	3	7	3	4	0	2	5	1	3	3
	7	0	0	2	9	6	0	3	7	7	2	4	3	13	3	1	0	1	10	9	0

Table E.4: Occurrences feature combinations of operator six

block		**shape**							**colour**				**orientation [°]**				**y-coordinate [-]**				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	4	2	2	5	8	6	1													
	g	2	2	6	4	2	4	0													
	b	8	4	3	4	6	2	3													
	y	3	4	0	3	1	0	3													
orientation [°]	0	17	1	2	13	10	6	0	15	11	16	7									
	90	0	3	2	3	7	6	2	10	4	9	0									
	180	0	7	5	0	0	0	2	2	5	3	4									
	270	0	1	2	0	0	0	3	1	0	2	3									
y-coordinate [-]	1	2	0	0	0	4	0	3	2	0	6	1	5	3	1	0					
	2	11	4	5	1	2	1	0	7	8	6	3	13	5	6	0					
	3	1	7	4	11	8	1	2	11	6	11	6	16	10	6	2					
	4	3	0	2	3	3	5	2	3	5	7	3	13	1	1	3					
	5	0	1	0	1	0	5	0	5	1	0	1	2	4	0	1					
x-coordinate [-]	1	5	3	0	0	2	2	0	4	1	5	2	7	5	0	0	0	7	4	1	0
	2	0	1	1	0	3	3	0	3	3	0	2	5	1	2	0	1	2	1	3	1
	3	2	2	5	0	3	5	0	5	4	5	3	6	6	5	0	0	6	5	2	4
	4	1	2	2	1	2	1	2	1	3	5	2	5	2	2	2	3	1	4	3	0
	5	3	4	1	1	3	0	1	2	2	8	1	6	1	4	2	3	1	6	2	1
	6	6	0	1	2	0	1	2	3	6	3	0	9	3	0	0	2	7	1	1	1
	7	0	0	1	12	4	0	2	10	1	4	4	11	5	1	2	0	0	13	6	0

Table E.5: Occurrences feature combinations of operator seven

block		shape							colour				orientation [°]				y-coordinate [-]				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	5	1	4	5	8	5	17													
	g	2	6	4	4	5	9	1													
	b	9	2	8	4	2	4	4													
	y	6	6	5	2	4	5	4													
orientation [°]	0	22	0	3	10	11	5	4	12	15	16	12									
	90	0	4	11	5	8	18	8	21	9	11	13									
	180	0	9	4	0	0	0	12	9	5	5	6									
	270	0	2	3	0	0	0	2	3	2	1	1									
y-coordinate [-]	1	1	2	1	2	8	3	5	4	7	4	7	8	6	7	1					
	2	6	7	7	4	1	3	9	12	9	8	8	11	14	11	1					
	3	8	3	10	5	7	1	6	12	4	11	13	18	18	3	1					
	4	7	3	2	1	3	8	4	9	9	8	2	12	9	4	3					
	5	0	0	1	3	0	8	2	8	2	2	2	6	7	0	1					
x-coordinate [-]	1	2	3	3	0	0	5	5	6	6	3	3	5	12	1	0	0	8	3	7	0
	2	2	3	1	3	4	4	5	12	4	3	3	6	12	3	1	4	6	1	3	8
	3	1	2	7	1	3	6	1	1	4	8	8	4	14	3	0	4	6	8	1	2
	4	5	2	2	1	2	4	2	1	5	5	7	10	3	2	3	1	3	8	5	1
	5	6	2	7	1	5	0	5	9	6	5	6	12	6	6	2	8	5	8	5	0
	6	6	3	0	1	1	4	7	9	3	6	4	11	2	9	0	5	5	5	4	3
	7	0	0	1	8	4	0	1	7	3	3	1	7	5	1	1	0	4	7	3	0

Table E.6: Occurrences feature combinations of operator eight

block		shape							colour				orientation [°]				y-coordinate [-]				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	4	3	0	4	1	2	8													
	g	0	4	2	4	1	2	2													
	b	2	3	2	2	2	2	1													
	y	1	7	1	1	2	1	3													
orientation [°]	0	7	7	1	4	1	2	2	10	5	5	4									
	90	0	0	2	7	5	5	3	5	5	7	5									
	180	0	7	1	0	0	0	8	5	4	1	6									
	270	0	3	1	0	0	0	1	2	1	1	1									
y-coordinate [-]	1	1	0	0	2	2	1	5	6	1	1	3	4	2	5	0					
	2	3	6	2	0	3	0	2	3	5	2	6	3	6	6	1					
	3	2	2	2	5	1	1	4	6	4	4	3	6	7	3	1					
	4	1	5	1	3	0	3	2	5	4	3	3	6	4	2	3					
	5	0	4	0	1	0	2	1	2	1	4	1	5	3	0	0					
x-coordinate [-]	1	2	0	1	0	0	1	0	2	1	0	1	2	2	0	0	0	1	2	1	0
	2	0	1	0	1	2	2	2	4	1	1	2	2	5	1	0	1	3	1	2	1
	3	2	3	2	3	0	2	1	2	3	5	3	3	6	2	2	1	2	4	4	2
	4	0	3	2	3	2	1	3	3	5	4	2	4	4	3	3	4	4	4	1	1
	5	1	2	0	1	1	0	4	5	2	1	1	2	2	5	0	4	1	2	1	1
	6	2	7	0	1	1	1	3	4	2	3	6	9	3	3	0	1	4	2	5	3
	7	0	1	0	2	0	0	1	2	1	0	1	2	0	2	0	0	1	2	1	0

Table E.7: Occurrences feature combinations of operator nine

block		shape							colour				orientation [°]				y-coordinate [-]				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	4	4	5	6	4	3	6													
	g	2	9	4	2	4	6	3													
	b	4	2	3	7	2	5	5													
	y	3	5	6	2	5	6	7													
orientation [°]	0	13	4	3	10	10	8	1	14	10	15	10									
	90	0	7	5	7	5	12	9	11	9	9	16									
	180	0	6	3	0	0	0	7	2	8	3	3									
	270	0	3	7	0	0	0	4	5	3	1	5									
y-coordinate [-]	1	2	1	2	0	6	3	4	3	4	2	9	7	8	2	1					
	2	8	9	6	2	3	1	4	11	8	6	8	10	12	6	5					
	3	1	3	4	9	3	1	7	5	5	11	7	14	7	5	2					
	4	2	5	4	5	3	8	3	10	8	8	4	13	11	3	3					
	5	0	2	2	1	0	7	3	3	5	1	6	5	7	0	3					
x-coordinate [-]	1	5	6	1	0	1	1	3	3	3	6	5	6	11	0	0	0	9	1	7	0
	2	0	1	1	0	2	12	3	4	2	5	8	6	12	1	0	4	3	3	3	6
	3	3	2	5	2	1	2	1	5	4	4	3	9	5	2	0	1	6	4	4	1
	4	2	4	7	2	6	1	5	8	8	3	8	11	3	2	11	7	6	4	5	5
	5	1	4	1	4	4	0	2	1	6	6	3	5	4	7	0	2	6	7	1	0
	6	2	2	1	1	0	4	4	4	5	1	4	9	4	1	0	4	2	1	4	3
	7	0	1	2	8	1	0	3	7	2	3	3	3	6	3	3	0	1	8	6	0

Table E.8: Occurrences feature combinations of operator eleven

block		shape							colour				orientation [°]				y-coordinate [-]				
		O	L	L (m)	Z	Z (m)	I	T	r	g	b	y	0	90	180	270	1	3	3	4	5
colour	r	11	3	4	4	8	5	9													
	g	1	9	7	4	6	5	7													
	b	5	7	5	8	9	8	2													
	y	5	4	5	5	6	4	7													
orientation [°]	0	22	4	6	9	17	6	5	23	10	18	18									
	90	0	4	5	12	12	16	8	14	16	20	7									
	180	0	8	8	0	0	0	9	6	6	3	10									
	270	0	7	2	0	0	0	3	1	7	3	1									
y-coordinate [-]	1	7	1	1	2	11	1	6	9	2	10	8	19	5	5	0					
	2	7	9	7	0	9	2	8	10	17	7	8	12	17	6	7					
	3	5	5	8	10	4	4	5	8	10	8	15	16	14	10	1					
	4	3	5	3	2	5	6	4	10	5	9	4	15	8	4	1					
	5	0	3	2	7	0	9	2	7	5	10	1	7	13	0	3					
x-coordinate [-]	1	5	3	1	0	2	2	2	2	1	5	7	8	7	0	0	0	6	3	6	0
	2	3	2	1	1	3	4	7	11	4	2	4	9	11	1	0	6	8	2	3	2
	3	2	4	6	3	3	10	2	2	3	18	7	9	15	5	1	2	5	8	5	10
	4	2	3	7	3	7	5	5	9	8	8	7	19	3	6	4	9	4	8	6	5
	5	4	4	3	4	5	0	3	7	7	4	5	9	5	6	3	9	2	6	2	4
	6	6	4	2	3	1	1	3	7	7	3	3	9	6	4	1	3	8	5	2	2
	7	0	3	1	7	8	0	3	6	9	4	3	6	10	3	3	0	9	9	4	0

Appendix F

Recreation of Previous Results

The work of Stellas [8] introduced a set of features of assembly instructions on which a decision tree was trained. These features are given in section 1.3.1. This decision tree outputs the target value, assembly time, based on these features' values. The only feature value that cannot be changed is the number of times an operator has assembled a product, so based on that fixated feature value, the other feature values are chosen so that the assembly time is minimal. Based on the chosen feature values, the corresponding assembly instruction is chosen. The reason for this recreation of results is to validate conclusions by Stellas [8] based on results of the actual setup, with a potential bigger sample size. This bigger sample sizes is never attempted to create, as is explained further in this chapter. The results in this chapter are based upon product $P_{F.1}$, as depicted in Figure F.1.

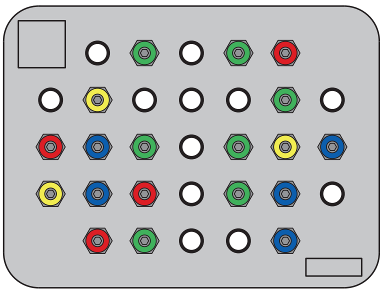


Figure F.1: Product $P_{F.1}$

Two example instruction sets are depicted in Figure F.3 and Figure F.4, which are very different based on the feature values introduced. Assembly instructions are introduced in section 2.3.3. As a first step, the assembly time data of all assemblies sorted by assembly instruction set ID are depicted in Figure F.2. The spread of the occurrences is depicted using boxplots, as introduced in section 2.5. The number of samples available for the feature value, is depicted using n = number of feature occurences.

Looking at this data, many notches overlap, which means that there is no significant difference between most of the assemblies based on assembly time. However, for example assembly id 85 versus 90 and 107

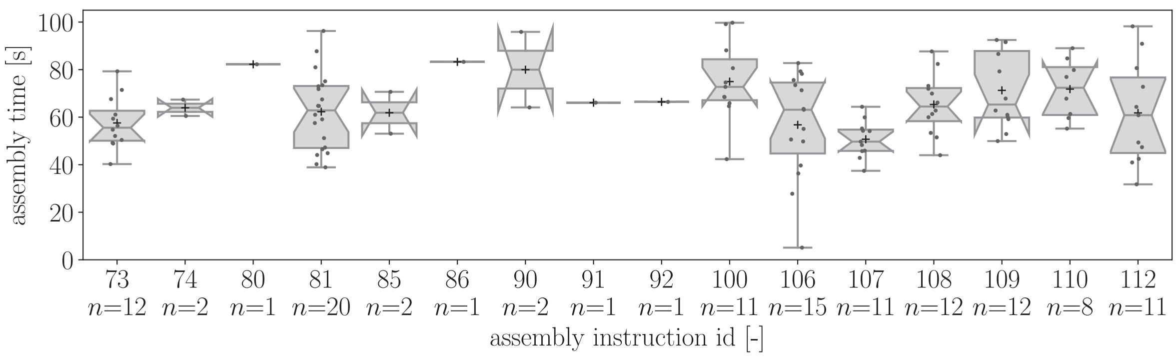
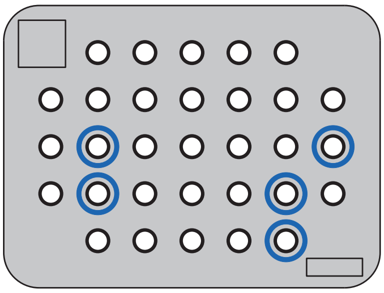
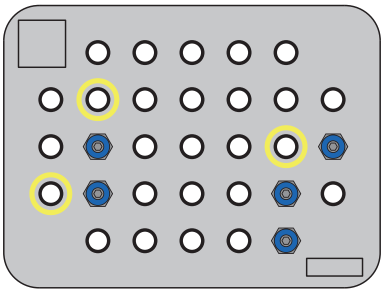


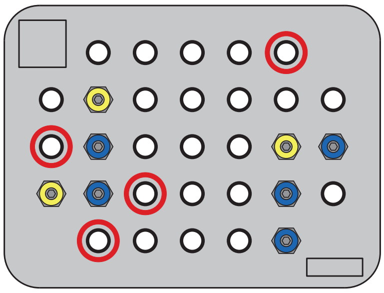
Figure F.2: Assembly time depicted against assemblies based on different assembly instruction sets



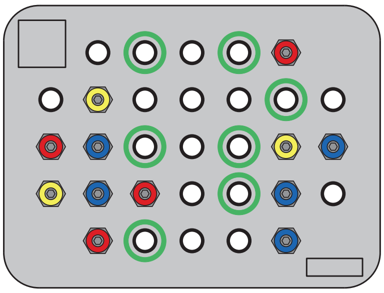
(a) Assembly instruction step one



(b) Assembly instruction step two

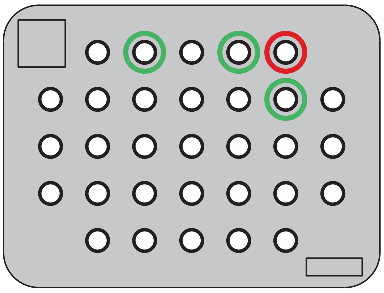


(c) Assembly instruction step three

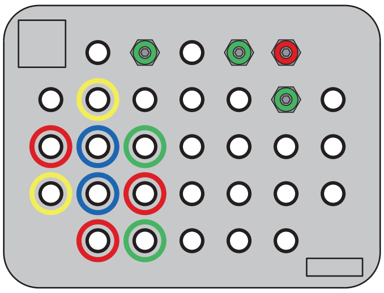


(d) Assembly instruction step four

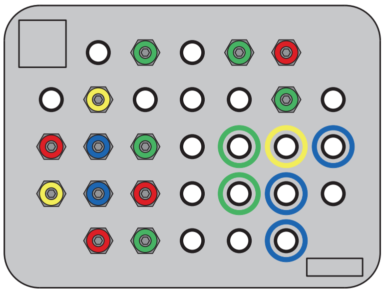
Figure F.3: Assembly instruction set 73



(a) Assembly instruction step one



(b) Assembly instruction step two



(c) Assembly instruction step three

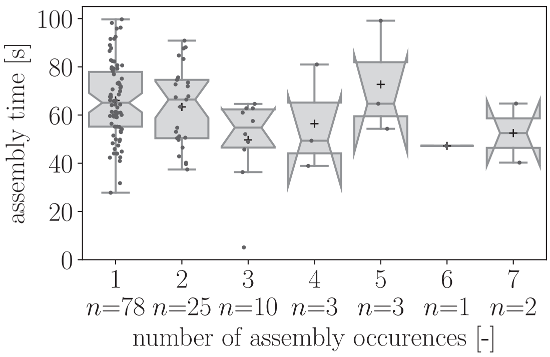
Figure F.4: Assembly instruction set 108

versus 108 do show significant differences. Note that data of all operators is taken into account, as no single operator assembled many products based on different instruction sets. Results are therefore limited, due to the shown performance difference between operators in section 6.3. Also note that no information is available about assembly conditions and operator behaviour, so all assemblies with an assembly time greater than 100 seconds are filtered out, based on manual inspection of the data. The data used is gathered during demonstrations of the setup, and are therefore black-box and not grey-box like the results based upon the experiments defined in section 5.5, as no notes were taken during these demonstrations. The overview of results presented here should therefore not be viewed as conclusive.

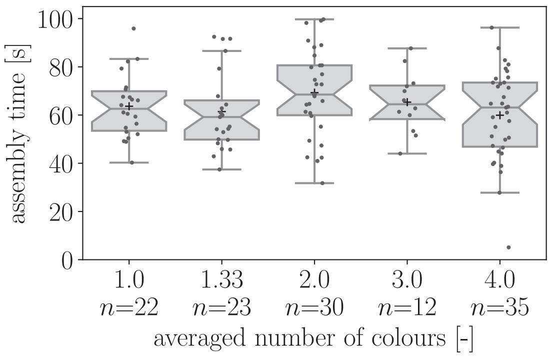
The assembly performance is first depicted against single features. Figure F.5a depicts assembly time against number of times an operator assembled the product. For the data with at least a sample size of five, so one, two or three occurrences, it is noticeable that the assembly time decreases over the number of assembly occurrences, but statistically the medians could still be equal. Note that the assembly time of a second occurrence, is not added to the first occurrence data, so while this product that has been assembled twice, is therefore also assembled once, it is not added to the occurrence one. Figure F.5b depicts the assembly time against the averaged number of colours per assembly step. Note that the horizontal axis is not to scale. No statistical differences are observed. Figure F.5c depicts the assembly time against standard deviation of the number of screws per assembly step, where again no differences are observed. Note that the horizontal axis is not to scale. Figure F.5d depicts assembly time against the averaged number of assembly steps in the assembly instruction. Finally, Figure F.5e depicts the assembly time against the averaged number of screws in the assembly instructions, where also no differences are observed. This is is unexpected, as the time needed by the setup to switch to the new instruction set step, was observed to slow down operators during demonstration of the setup. This could indicate that the inclusion of multiple operators with varying overall assembly times overshadows patterns in the data depicted.

As the results in section 6.6.2 have shown, analysing data based on single features is not useful. Therefore, Figure F.6 depicts the assembly time versus the standard deviation of the screws over the assembly steps, with hues representing the average number of colours over the assembly steps. No statistical differences are observed. Figure F.7 depicts the assembly time versus the number of instruction set steps, with hues representing the standard deviation of the screws over the assembly steps. No statistical differences are observed. Figure F.8 depicts the assembly time versus the number of instruction set steps, with hues representing the number of times an operator assembled that product that day. No statistical differences are observed.

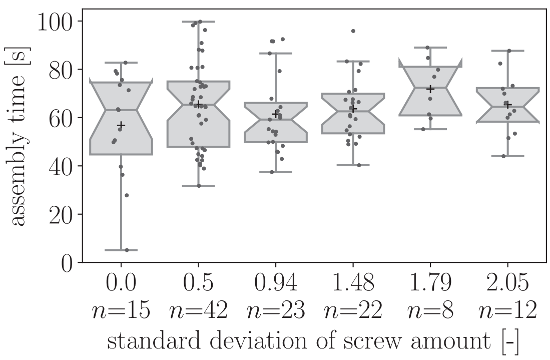
Overall, the results Stellas presented are applicable in this recreation of his results too. The dataset is different and black-box, and while the results clearly show that this is not ideal as no information is present concerning assembly conditions, the only feature that seems to influence assembly time in some degree is the number of times an operator has assembled a product, which is not a feature that influences the instruction sets itself. Recreation of Stellas results with the experiment data as presented in section 5.5 is not possible, as instruction sets are not used. In addition to these observations, the analysis by Stellas requires a substantial number of sample occurrences of different assembly instruction sets of the same product, which does not align with this research's focus on high-mix and low-volume production. Based on these results, other product features are introduced in section 4.5



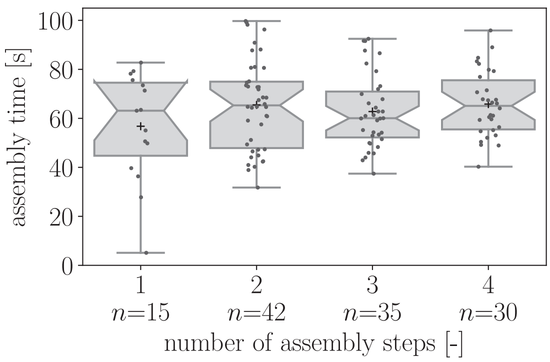
(a) Assembly time depicted against number of times an operator assembled the product



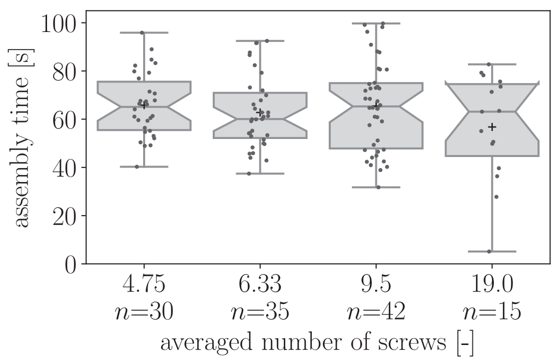
(b) Assembly time depicted against averaged number of colours per assembly step



(c) Assembly time depicted against standard deviation number of screws per assembly step



(d) Assembly time depicted against averaged number of assembly steps in assembly instructions



(e) Assembly time depicted against averaged number of screws in assembly instructions

Figure F.5: Assembly performance depicted against features from Stellas [8]

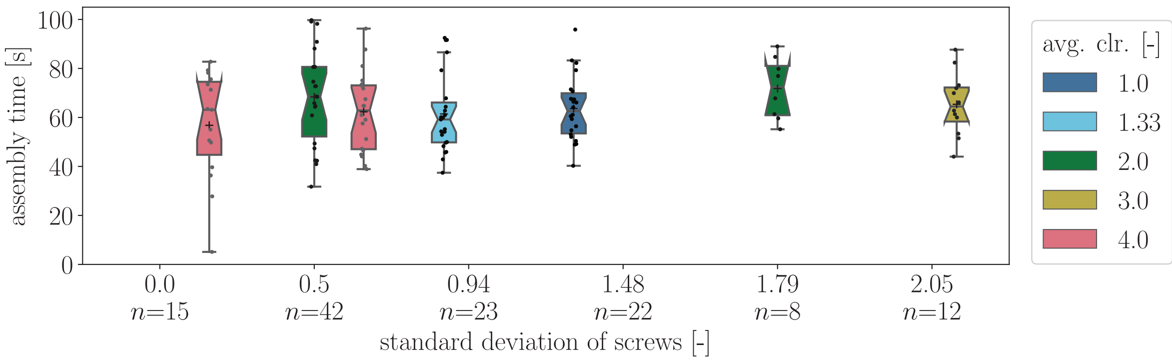


Figure F.6: Assembly time depicted against standard deviation of number of screws per assembly step, hues depict average number of colours per assembly step

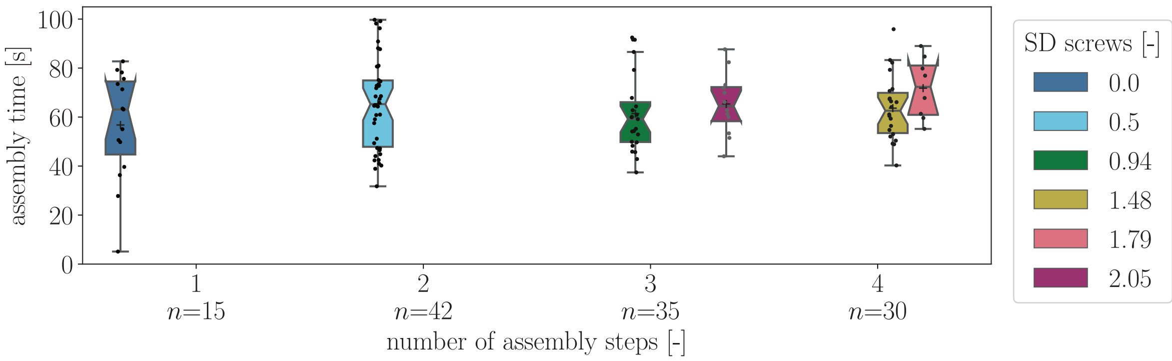


Figure F.7: Assembly time depicted against number of assembly steps in instructions, hues depict standard deviation of number of screws per assembly step

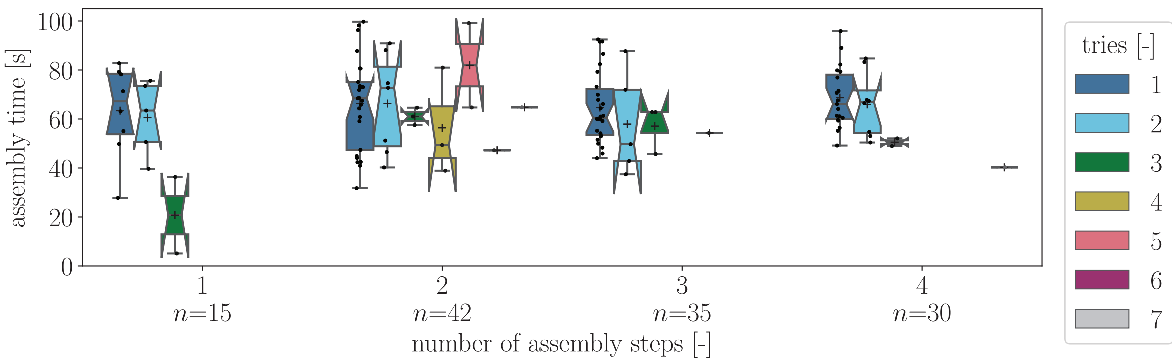


Figure F.8: Assembly performance depicted against number of assembly steps in instructions, hues depict number of assembly occurrences

PO Box 513
5600 MB Eindhoven
The Netherlands
tue.nl

DEPARTMENT OF MECHANICAL ENGINEERING

EINDHOVEN
UNIVERSITY OF
TECHNOLOGY