DEVELOPMENT OF AUTONOMOUS COLLABORATIVE MULTI-ROBOTICS ASSEMBLY SYSTEM WITH DIGITAL TWIN APPROACH

by

Kyaw Htet Oo

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Industrial and Manufacturing Engineering

Examination Committee:	Dr. Pisut Koomsap (Chairperson) Prof. Huynh Trung Luong Dr. Mongkol Ekpanyapong
Nationality:	Myanmar
Previous Degree:	Bachelor of Science in Engineering in Mechatronics Engineering Asian Institute of Technology, Pathumthani, Thailand
Scholarship Donor:	AIT Fellowship

Asian Institute of Technology
School of Engineering and Technology
Thailand
December 2022

AUTHOR'S DECLARATION

I, Kyaw Htet Oo, declare that the research work carried out for this thesis was in accordance with the regulations of the Asian Institute of Technology. The work presented in it are my own and has been generated by me as the result of my own original research, and if external sources were used, such sources have been cited. It is original and has not been submitted to any other institution to obtain another degree or qualification. This is a true copy of the thesis, including final revisions.

Date:

Name (in printed letters): Kyaw Htet Oo

Signature: 

ACKNOWLEDGMENTS

First and foremost, I want to express my great thanks to Dr. Pisut Koomsap, my thesis supervisor, for allowing me the chance to complete my thesis and for his important advice and patience during the process. Being able to work and study under his direction was a huge honor and privilege.

My sincere gratitude also goes to the committee members Dr. Huynh Trung Luong and Dr. Mongkol Ekpanyapong who generated widespread interest in my thesis work and helped me along the way by offering advice and comments that encouraged and pushed me to carry on with my studies.

I will always be appreciative of the financial assistance I received from Erasmus and the EU Program. I would not have been able to finish this thesis without the funding the program provided. I am incredibly grateful to my friends in the A-cube research group for their invaluable support and contributions to the completion of this project.

Additionally, I would like to express my gratitude to Mr. Thanit, the ISE laboratory supervisor, and Mr. Manh, the research specialist, who both provided me with technical support and allowed me to utilize all of the lab's machinery and tools.

I want to express my sincere gratitude to my family, especially my parents, for their love and support—both materially and, most importantly, psychologically. Indeed, without them, none of my accomplishments would be conceivable.

Last but not least, I want to extend a particular thank you to my partner Phoo Moh Moh Thein, who has been there for me over these years and has consistently provided understanding, encouragement, and support.

I am still overwhelmed by the kindness shown to me throughout the writing of my thesis, and I am unable to explain how much I appreciate the physical and mental assistance I have gotten in words.

ABSTRACT

In todays' manufacturing sector, Robotics Assembly (RA) has been broadly used as an important role technology. In RA, robots use the most efficient processes to produce a variety of products ranging from large-scale products to minute products. A number of tasks, including part positioning, mating, adjusting, and assembling parts or components to produce finished products or partially assembled parts, can be carried out by assembly robots. However, due to multiple sources of uncertainty including fixtures, end-effectors, and actuators, completing assembly tasks by assembly robots continues to be a significant challenge. Identifying the best search approach to increase positioning accuracy during assembly is the most significant technical obstacle. The 3D robotics assembly system arranges the assembly order step by step, according to information from a final assembly configuration. Robotics assembly planning is a critical process, and it is done for each assembly process in two steps: planning the pre-assembly process and planning the assembly process itself. There is no issue regarding the second step where all assembly parts are reassembled in order according to the final assembly configuration. Implementing the first step is essential, though, as it directly affects how difficult the assembly system becomes when it starts to handle a random arrangement of complex 3D parts during initial setup. When the initial part configurations are considered to be random configurations without fixtures, the shape of the assembly part, the initial position of the assembly part and the initial stable orientation of each assembly part are three key factors to be considered for planning the pre-assembly process. Furthermore, the problem becomes more complex when the used assembly parts are in complex shape with random stable orientation, and they must be converted to the final assembly configuration from any random initial configuration. Some researchers conducted this area with the use of simple 3D objects for instance, soma blocks and Lego blocks which means still far to reach the real-life level industrial assembly applications. Therefore, the main idea of this study becomes to reach real-life industrial assembly applications by using complex shape 3D parts in random initial configuration setup. This research aims to improve the 3D assembly process of Co-MRAS to better support the handling of complex 3D parts in random configurations at the initial state.

CONTENTS

LIST OF TABLES

Tables **Page**

LIST OF FIGURES

Figures **Page**

Figures **Page**

LIST OF ABBREVIATIONS

RA	Robotics Assembly
RRC	Robot-Robot Collaboration
MRAS	Multi-Robotics Assembly System
Co-MRAS	Collaborative Multi-Robotics Assembly System
PiH	Peg in Hole
DT	Digital Twin
Co-RA	Collaborative Robotics Assembly
HRI	Human Robot Interaction
HMI	Human Machine Interaction
HRC	Human Robot Collaboration
MRAPS	Multi-Robots Assembly Planning System
AI	Artificial Intelligence
ML	Machine Learning
CNN	Convolutional Neural Network
RCNN	Region based Convolutional Neural Network
Mask-RCNN	Mask Region based Convolutional Network
DL	Deep Learning
3DA	Three-Dimensional Assembly

CHAPTER 1
INTRODUCTION

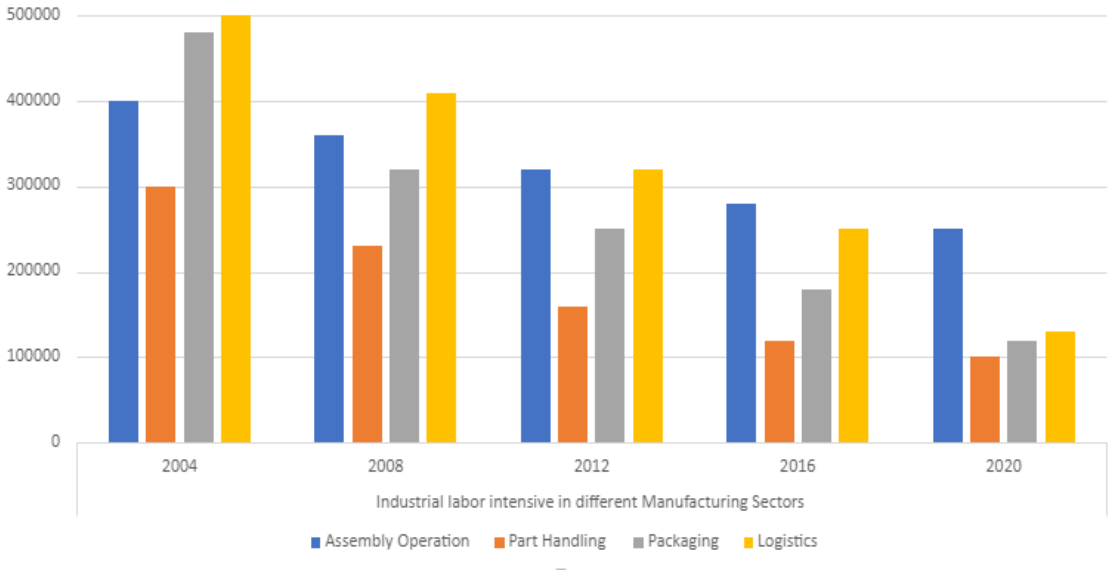
1.1 Background of the Study

As a manufacturing process, assembly refers to the sequential accumulation of components, sub-assemblies and software that produces the usable final products. Industrial assembly could be distinguished from non-repetitive or hobby-assembly by focusing on productivity, efficiency and cost-effectiveness. Assembly process differs from other conventional manufacturing processes in a few characteristics such as a large number of variations, handy grasping of assembly components and frequent changes in the production system.

These qualities necessitate flexibility in the assembly process to accommodate its potential and efficiency. Humans have always been regarded as the best resource for flexibility relating to the assembly. Therefore, compared to other industrial fields, the impact of widespread automation on assembly is substantially lower. These days, assembly is now regarded as the most labor-intensive procedure in the production of discrete products. (See Figure 1.1).

Figure 1.1

Labor Intensive in Industrial Assembly (Source: Industrial Labor Report, 2021)



The best solution for the assembly operation is no assembly process. Although additive manufacturing (AM) technology is widespread and has the potential to eliminate some joint assembly processes, the elimination of the entire assembly process cannot be achieved with AM technologies alone. As a result of increasing market challenges, the degree of automation of manual assembly cells to robotic assembly cells is rapidly increasing. (See Figure 1.2). Additionally, Figure 1.3 makes illustration of the use of assembly in science research demonstrates how interest in robotics technology has grown over time. It has been noted that since 2003, the number of publications has dramatically increased.

Figure 1.2

Annual Installations of Industrial Robots (Source: World Robotics, 2020)

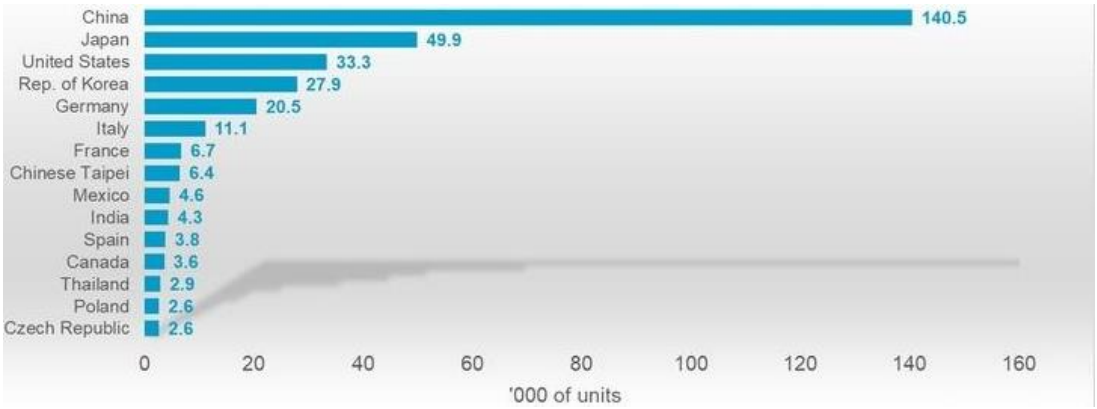
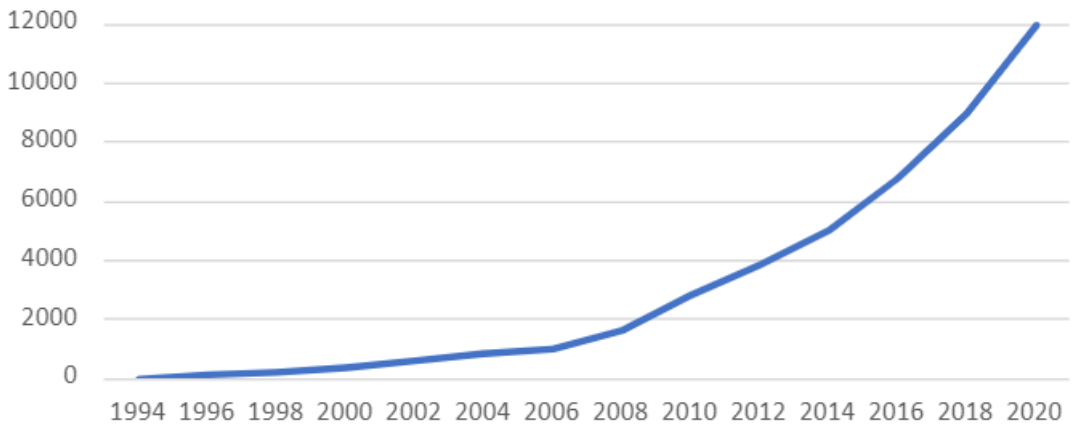


Figure 1.3

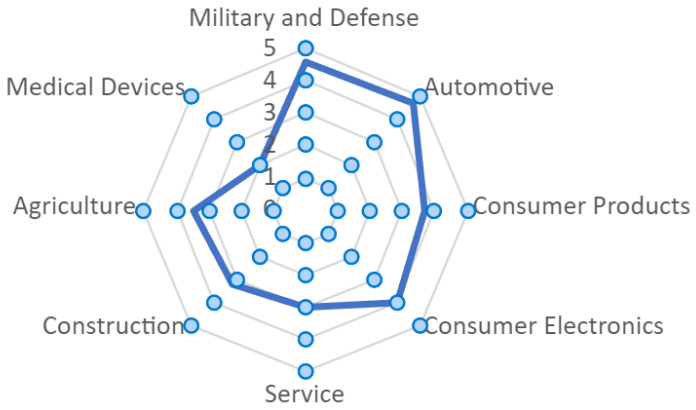
Robotics Applications in Scientific Research Papers (Source: World Robotics, 2019)



Consumer electronics, military applications and defense, automotive and medical robots are the most developed industries, and Figure 1.4 illustrates how forecasting of the adoption rate of robots differs dramatically between industries (World Robotics, 2020). The assembly line production system developed naturally to the robotics assembly system, where robots now perform the specialized tasks that were previously completed by human labor (Chryssolouris et al., 2009).

Figure 1.4

Robotics Adoption Rates Across Industries: 2015–2025 Projections (Source: 2018 World Robotics)



In traditional assembly line system, the assembly system that provides for mass production and speeds up production compared to single assembly station, but there is no flexibility for customization because doing so would be expensive and time-consuming. However, as technology developed, it became evident that the robotics assembly cell, in which a single station is responsible for producing a finished product or a sub-assembled component, can improve production performance and boost agility to deal with product variability and customization (Kousi et al., 2019).

In the assembly work cell, a single robot typically completes a straightforward repeating activity with minimized positional uncertainty (Michalos et al., 2010). However, the kinds and numbers of assemblies that can be finished with these single manipulator robots' technologies for assembly are constrained. Such single robotics assembly systems are inherently rigid since they depend on expensive fixtures or externally

integrated axes, and any corresponding fixturing must be completely reconfigured whenever a significant change in the assembly process occurs.

The degree of automation of single robot assembly cells, however, is greatly expanding thanks to human-machine collaboration as a result of growing market challenges. A new class of industrial robots known as collaborative robots, or cobots, is now making the fantasy of humans and machines cooperating toward a common objective reality. Collaborative robots should be lightweight, simple to program, and less dangerous to people.

1.2 Statement of the Problem

Implementing Human-Robot Collaboration (HRC) in industrial production can help achieve multiple goals. According to various studies, the benefits of increasing operational efficiency, i.e., cheap unit production cost and increased production rate, are the most desired goals of having HRC. However, the global pandemic Covid-19 resulted in a lack of industrial employees and had a tremendous detrimental impact on the workforce, and even the HRC cannot adjust in a pandemic situation.

By late December, the World Health Organization reported that COVID-19, the illness brought on by the coronavirus SARS-Cov-2, had afflicted more than 79.2 million individuals globally (World Health Organization, 2020). As a result, by bringing several robots instead of human workers to divide the burden and tasks, RRC robot-robot-collaboration should be able to adjust to the pandemic situation and continue conducting production.

Such robot-robot collaborations (RRC) in assembly systems are intrinsically more difficult to program and integrate initially and require advanced sensing and control capabilities to enable the complete assembly algorithms. A very few studies approached this topic. The First One is “Multiple mobile robots planning for Furniture Assembly: Chair Assembly” (Dogar et al., 2019) found the assembly sequence for several mobile robots to assemble a chair. The method considered the constraints between different robots, between the robot and assembly components, and between mating components and assembled components. This multi-robot assembly system addresses the problem of finding robot configurations to grasp assembly parts during a sequence of collaborative assembly operations.

The second article "Multi robot assembly planning system, Chair Assembly" (Yu et.al, 2021), the authors construct a robotics assembly simulation environment in which two robots must sequentially pick up two parts, plan collision-free motion trajectories, precisely mate two parts, and then return to the previous pickup movement. The 220 chairs from the PartNet dataset were utilized in this investigation. The relative postures of the parts were altered to satisfy collision and contact criteria, and connection points and graspable regions were annotated.

In a different study titled, "Dual Arm Assembly Planning for Soma Block Assembly," by (Chen et al., 2021) describes a planner that can automatically determine the best assembly order for a dual-arm robot to assemble the soma blocks. The planner generates all potential assembly sequences using the mesh model of the objects and the assembled state, and then evaluates the best assembly sequence by taking stability, graspability, assemblability, and the requirement for a second arm into account. The system presents a way that uses a second arm to handle the unstable components, which is the key improvement. It also develops a more reliable stability analyzer.

Moreover, the current workforce is suffering from an increase in an aging society. In the near future, skilled human workers who are old and cannot handle the task physically but still can support their technical skill virtually. However, digital representation of the production environment and process has developed as a way to handle some of the production system performance optimization (Kousi et al., 2018).

The benefits that the Digital Twin technology may provide in terms of giving perception and cognitive capacities towards more autonomous and intelligent robotic systems have made it a hot topic in this era of digitalization in manufacturing (Chryssolouris et al., 2009). In current multi-robotics assembly research areas, flexibility and adaptability of the pre-assembly planning processes are still constrained due to the rigid use of fixed known position and orientation at the initial state (Michalos et al., 2016), and the use of simple shape assembly part models (Makris et al., 2012).

However, existing applications have limited perception capabilities not allowing them to handle complex shape parts in assembly operation and not allowing robots to adapt in dynamic random part configuration at the initial state. Most of the manipulators are restricted to performing off-line programmed tasks only when simple shape assembly

work pieces are in fixed positions with rigid, expensive fixtures. To overcome the existing limitations in multi robotics assembly system, this paper contributes a Digital Twin approach able to perform collaborative multi robotics assembly tasks for complex shape parts in 3D space without fixtures.

1.3 Objectives of the Study

The main objective of this research is to develop an autonomous collaborative multi-robotics assembly system with a digital twin approach to handle the complex shape of 3D objects in initial random configurations.

1.4 Limitations and Scopes

1. In this study, initial grasps and final grasps are needed to be confirmed by the operator in the virtual world.
2. Optimal paths planning for collision avoidance between multiple robotics arms are not focused in this study.
3. The distortion errors caused by camera resolution are only constrained by feasible area.
4. Only 2D images are analyzed to extract 3D part's coordinates.
5. Only two collaborative robots will be used due to the limited resources available.

CHAPTER 2

LITERATURE REVIEW

The technologies utilized in assembly robots are described in this chapter along with an overview of the various stages and types of robotic assembly. As multi robotics assembly system is an ordinary addition of robotics assembly technology, the current issues such as task allocation, collision avoidance for multiple robotics arms and time management for multiple robotics arms are presented in this chapter. Moreover, how collaboration and digital twin technology comes into play in multi-robotics assembly and current state of robot-robot collaboration (RRC) are discussed for managing the multiple robotics arm to eliminate the use of rigid fixtures and increase flexibility in modern production line.

2.1 Robotics Assembly (RA)

Robotic assembly (RA) has become a common practice in the manufacturing sector. RA offers numerous advantages over the competition with manual assembly and automatic assembly. For instance, RA is flexible to change in product design and production system. Thus, if the product design or production layout changes, the robot can be reprogrammed to adapt the new product design or production layout.

A RA system is additionally capable of handling a far higher output volume as compared to a high-speed automatic assembly machine. Multiple product variations can be handled by a RA system. For instance, a robot can be told to only select particular components for the finished assembly. Assembly robots can position, match, fit, and assemble product parts thanks to their extensive range of capabilities (Baykal, 2018).

Because of multiple cradles of uncertainty, for instance rigid fixtures, end-effectors, or different actuators, completing exact assembly tasks is still very difficult for robots. The most important technical difficulty is determining the optimum search technique to increase positioning precision when assembling. The following sections 2.1.1 and 2.1.2 provide a summary and discussion of the bulk of the robotics assembly types and process overview.

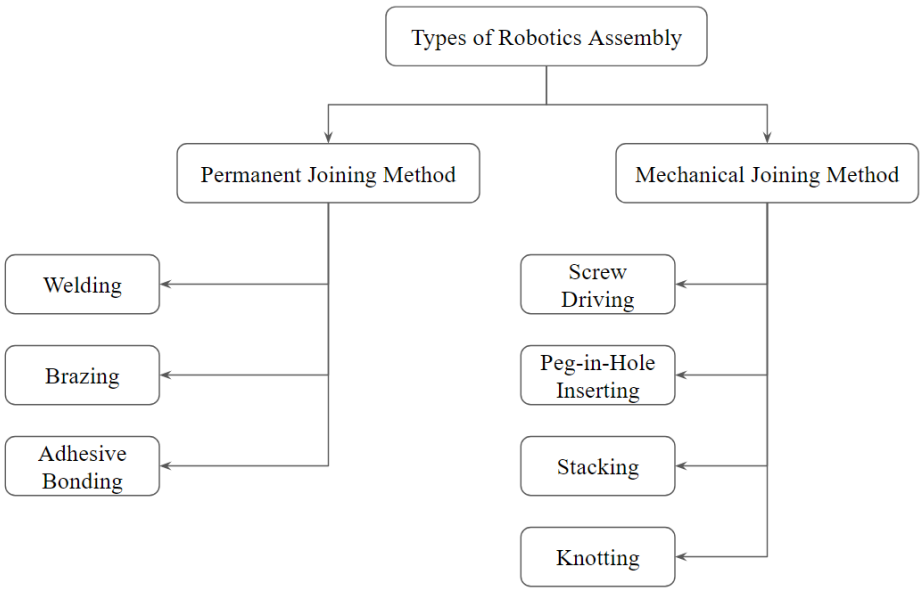
2.1.1 Robotics Assembly Types Overview

There are dozens of different robotics assembly processes in manufacturing and can be divided two main categories as Permanent joining methods and Mechanical joining methods as presented in Figure 2.1. The permanent joining process is where the assembly operation is, more or less, non-reversible, for example, the process like welding, brazing and adhesive bonding are included in permanent joining process.

Mechanical joining process refers the operation is, more or less, reversible which include, screw driving, peg-in-hole inserting, stacking and knotting. This work focuses on peg-in-hole assembly process, for this reason the other assembly processes are presented but not explained here.

Figure 2.1

Classification of Robotics Assembly Types Depending on Joining Methods



2.1.2 3D Robotics Assembly Stages: Peg-in-Hole Assembly

Industrial robots are utilized often in the production sector nowadays, particularly for the assembly of 3D items with pegs and holes (Da Xu, 2020). The generic 3D peg-in-hole RA process includes three main steps and the steps are presented as below.

Step 1: Identification

The generic modern RA process starts with part information collection and identification. There are many ways to identify and validate parts or components on

shape, size, color, barcode, QR code, and other features by using different types of machine visions algorithms according to the used of parts in assembly process (Nagarajan et al., 2016).

Step 2: Picking

The components that are already identified by the system are then picked up by robotic grippers and moved to the intended locations for assembly (Chang et al., 2017). Robotics assemblies can perform surface mount assembly and 3D assembly, depending on the degrees of freedom of the gripper and different assembly applications.

Step 3: Mating

After picking the part from the certain location, the picked parts are needed to place to a specific position and orientation on the worktable with certain fixture or another robotics arm to perform the joining or mate process. (Mei et al., 2015) The part mating process is done by joining two or more assembly components step by step until it reaches the final assembly configuration or final product. The part picking and placing or mating process will continue step by step until the final product is acquired.

2.1.3 Robotics Assembly Planning Strategies: Peg-in-Hole Assembly

The success rate of assembly operations is anticipated to increase with further planning strategy development (Xu et al., 2014). Human interaction is projected to decrease, prices to decrease, and high-precision assemblies will eventually be fully automated. A planning method is dependent on the availability of sensor data. Therefore, the key determinants of the resilience, constancy, and planning search accuracy in robotic assembly are the quantity and variety of sensors, data manipulating algorithms, and the incorporation of sensors and planning methodologies.

Early industrial robots could only perform a single, repeatable automated peg-in-hole assembly planning task with a single sensor due to limitations in budget, equipment, application, and sensor expansion. To put it another way, researchers suggested a peg-in-hole assembly planning method based on force sensors to emulate a person without the benefit of vision, relying solely on touch to fulfill the peg-in-hole assembly planning assignment for a certain variety of peg and hole components, and in order to replicate a human without the use of tactile sensations and perform the peg-in-hole assembly

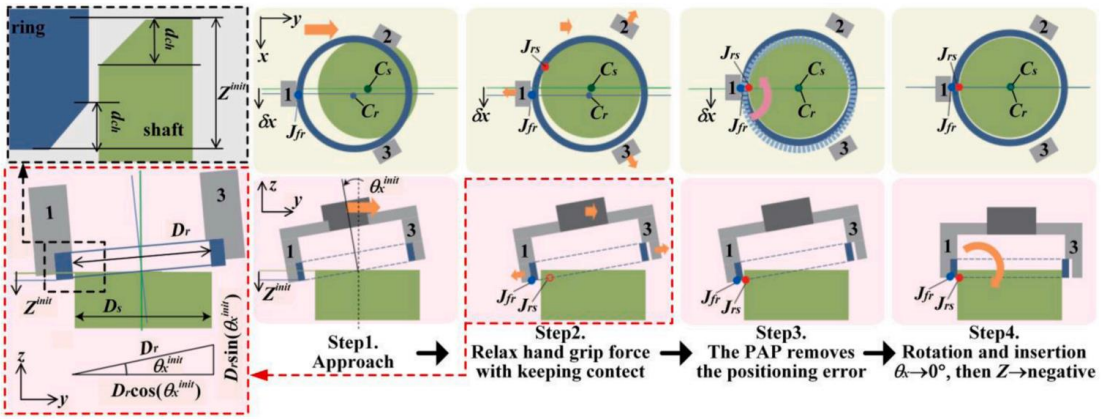
search job, researchers presented a visual sensor-based PiH assembly search technique (Yang et al., 2018)

The planning search path is pre-defined in the planning search algorithm in the force sensor-based peg-in-hole assembly search strategy (see Figure 2.2), and a threshold value is established for the $\frac{F}{T}$ generated during the peg and hole contact, and when the assembly robot is unable to meet the accuracy of autonomous assembly because of significant errors during positioning and repetition, the peg-in-hole assembly search task is accomplished using the feedback control of the $\frac{F}{T}$ sensor (Jiansong et al., 2018).

When the assembly robot is unable to meet the accuracy of autonomous assembly due to significant position arranging and repetition errors, the force sensor-based peg-in-hole assembly search strategy (see Figure 2.2) is used to complete the search task. A threshold value is selected for the force/torque produced during the peg and hole interaction, and the search path is pre-defined in the planning algorithm.

Figure 2.2

In the Process of Seeking PiH Assembly: Contact State Recognition and the Search Algorithm

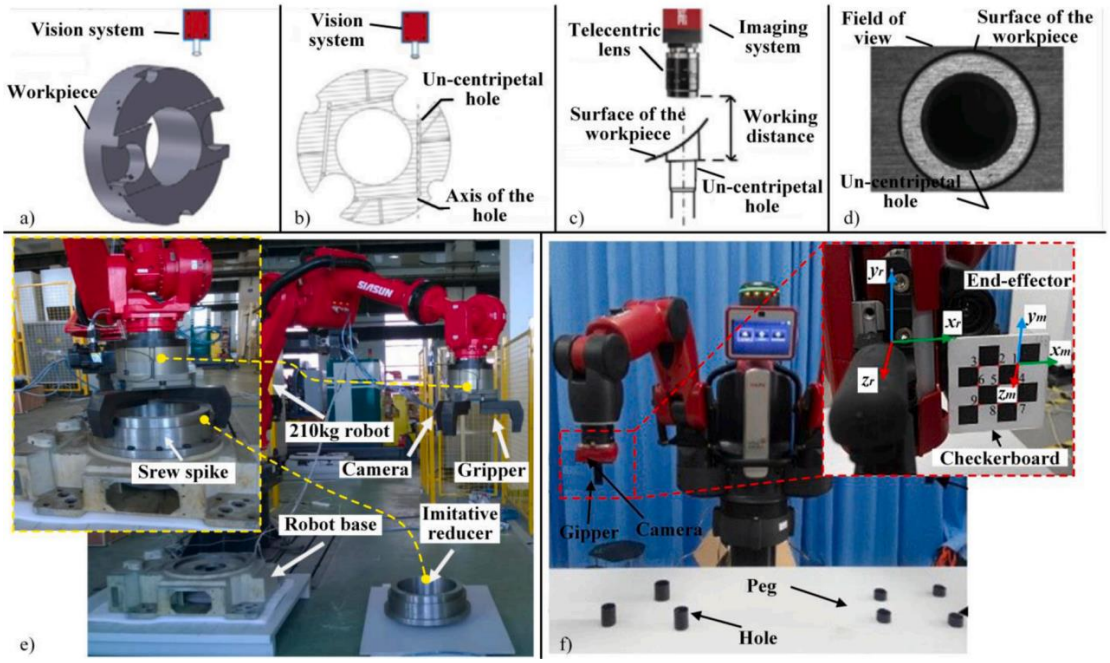


Even though they can complete a broad range of peg-in-hole assembly planning activities and also prevent the peg and hole components contact search, the high-accuracy peg-in-hole assembly planning strategy based on vision sensors is constrained by the camera distortion, environmental conditions, and sheltering in the vision system (see in Figure 2.3).

However, the laser intervention supervision planning strategy itself has a significant drawback: it necessitates the preliminary installation of the laser receiver on the completed component (Chang et al., 2017). In the assembly of larger peg and hole components, laser interferometric guiding is commonly utilized and for the peg-in-hole assembly planning problem, avoids the shortcomings of $^{F}/_{T}$ sensor-based and visual sensor-based systems.

Figure 2.3

Using a Vision Sensor as a Search Method for PiH Assemblies



2.1.4 Challenges of Search Strategies in Robotics Assembly

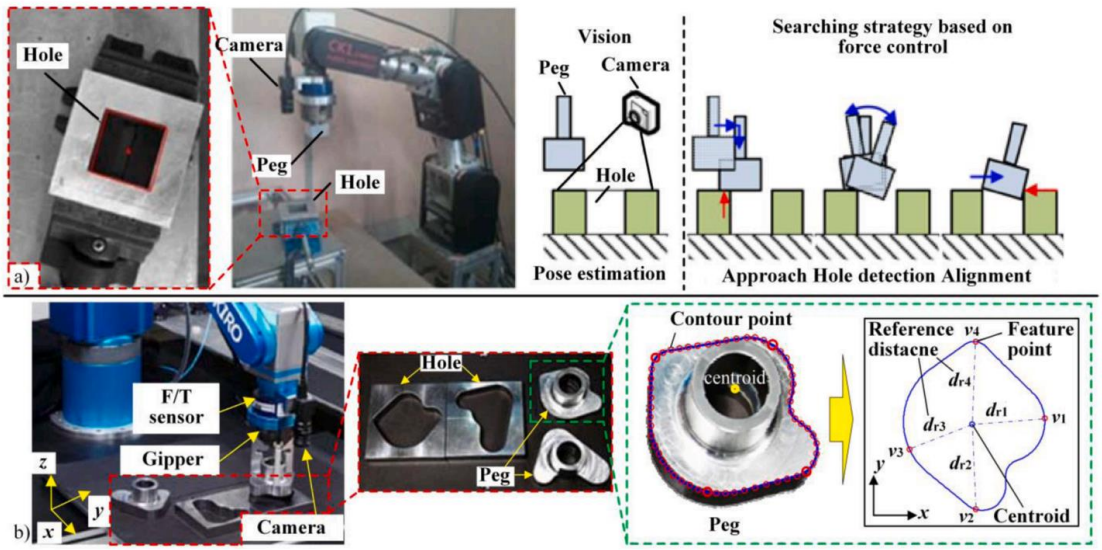
As time goes on, the peg-in-hole 3D assembly planning task is no longer restricted to a single straightforward task. Instead, it must now take into account the complexity of the assembled components, precision, and the improvement of fully autonomous assembly, which increases the demands on the peg-in-hole 3D assembly planning strategy. Simultaneously, high-speed computational ability expansion is also widely utilized in the assembly robotics planning process, of which the multiple sensors cooperation technique is one of the most common (see in Figure 2.4).

A two-stage planning algorithm is used during the peg-in-hole search process, realizing a wide range, complete autonomy, and high accuracy peg-in-hole assembly planning

task from visual search to contact interaction alignment (Zhang et al., 2013). In peg-in-hole assembly search tasks, collaboration with $^{Force}/_{Torque}$ and visual feedback sensors can overcome the boundaries of both categories of sensors to the highest extent possible, while also optimizing the existing advantages.

Figure 2.4

Usage of Vision and Force Sensors in the PiH Assembly Planning Strategy



Additionally, the coordination between operational controllers and anthropomorphic behavior and other positional sensed based actuators has substantially advanced peg-in-hole assembly planning strategies. However, the main drawbacks of multiple sensor-based coordination include in elevation price and difficult planning strategies (Sun et al., 2016). Multiple sensors-based cooperation combined with high level control algorithms to achieve assembly plan for peg-in-hole functional parts like complex shape polygons or complex shape curves has emerged as a significant movement as a result of advancements in sensor-based technology, better control and planning algorithms, and the ongoing development of AI and ML areas.

As a result, this analysis concluded that peg-in-hole assembly planning methodologies are crucial for resolving autonomous assembly issues and advancing robotics. There are researchers that have evaluated the peg-in-hole assembly search and planning strategies for robotics manipulators with simple 3D mechanical components currently, but they have not done a systematic and focused evaluation for the peg-in-hole 3D

assembly search and planning strategies for multi-robotics arms with complex shape including polygon curves in 3D parts.

2.2 Collaboration in Robotics Assembly (Co-RA)

Today's manufacturing business places a high value on collaboration, and robotic assembly which has been labelled the manufacturing industry of the future shows enormous potential for merging these two fundamental technologies. Traditionally, robust fixturing in the assembly work cell reduces positional uncertainty when a single robot completes a straightforward repeating activity. However, the kinds and numbers of assemblies that can be finished with these single manipulator robots' technologies for assembly are constrained (Tao & Hu, 2017).

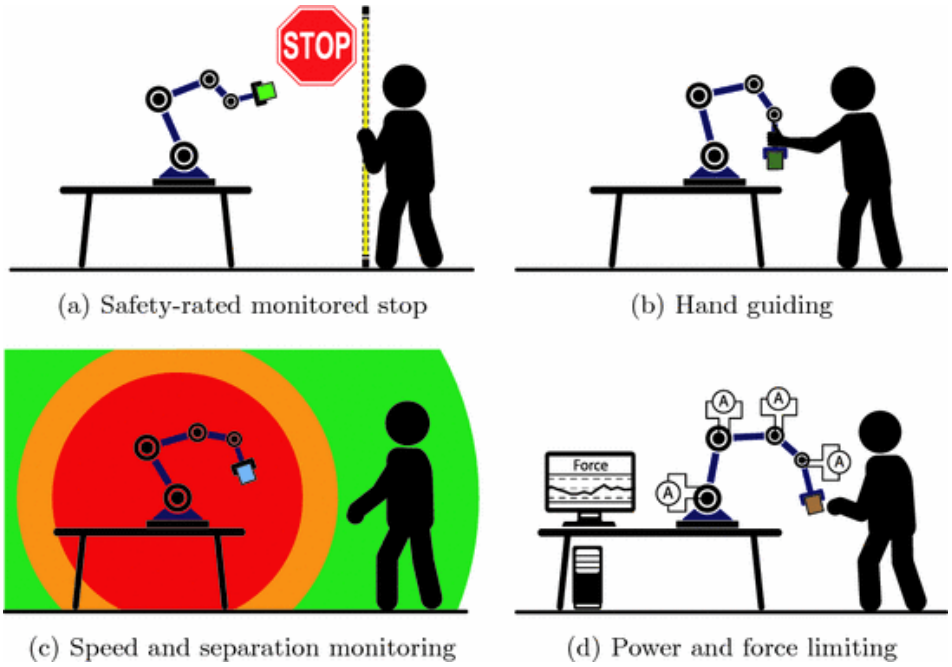
Due to their reliance on pricey fixtures or externally integrated axes, these single robotic assembly systems are inherently inflexible, and any associated fixturing must be entirely changed whenever a significant change in the assembly process takes place. Parts can be skillfully moved and relocated to finish more difficult assembly procedures by utilizing human robot cooperation (HRC) or additional robots in the assembly process as dynamic fixtures. Multiple robots can be used to provide assembly solutions that are more flexible and can easily adapt to changes in manufacturing procedures.

2.2.1 Human Robot Collaboration (HRC) in Robotics Assembly

Modern industrial robotics is transforming many industrial sectors throughout the world, and in so-called collaborative systems, operators are increasingly working closely with robotic agents. Cobots, also known as collaborative robots, have been shown to be able to improve working conditions and make tasks easier while reducing negative aspects of employment (Wang & Cho, 2008). The requirement to evaluate the operators' subjective mental effort and attention has grown vital due to contemporary improvements in human-machine systems, which got more sophisticated and automated (Mei et al., 2015).

Figure 2.5

Different Safety Interaction for Human Robot Collaboration (HRC).



(a) Safety-rated monitored stop (b) Hand guiding

(c) Speed and separation monitoring (d) Power and force limiting

The Human Robot Interaction (HRI) system must be properly constructed to minimize dangers to the human agent (See in Figure 2.5). Safety is a major concern in industrial robotic systems, and psychological factors like cognitive workload have a significant impact (Krüger et al., 2017). According to some authors, visual attention can be used as a measure of cognitive workload, and the two concepts are ultimately linked to safety results (Jiansong et al., 2018). Human-robot collaboration (HRC) has been shown to boost productivity and give businesses more flexibility (Michalos et al., 2016).

Robots could assist mitigate bad aspects of work and boost favorable ones, such as enhancing working conditions and simplifying jobs (Thomas et al., 2001). However, the global pandemic Covid-19 resulted in a lack of industrial employees and had a tremendous detrimental impact on the workforce, and even the HRC cannot adjust in a pandemic situation. By late December, the World Health Organization reported that COVID-19, the life-threatening disease brought on by the coronavirus SARS-Cov-2, had afflicted more than 79.2 million individuals globally (World Health Organization, 2020). Moreover, the current workforce is suffering from an increase in an aging society. In the near future, skilled human workers who are physically old and cannot handle the task physically.

2.2.2 Digital Twin (DT) for Robot-Robot Collaboration (RRC)

As a result, by bringing several robots instead of human workers to divide the burden and tasks, robot-robot-collaboration (RRC) should be able to adjust to the pandemic situation and continue conducting production without the shortage of human workers (Yu et.al, 2021). In order to enable the full assembly algorithms, such collaborative multi robots' 3D assembly require advanced sensing and control capabilities, which are inherently more complex to program and integrate at first.

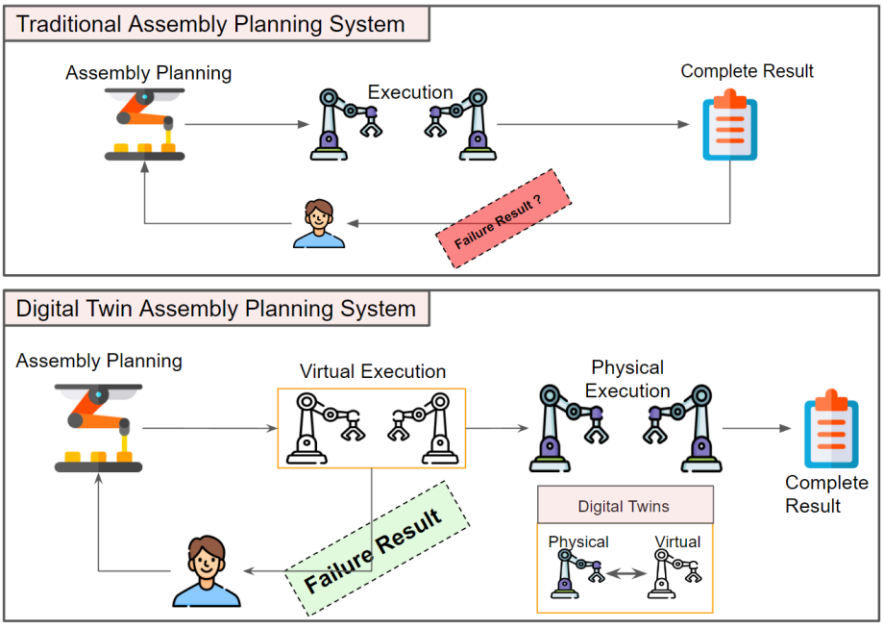
The multi-robotics assembly system arranges the assembly order step by step, according to information from a final assembly configuration. The robotics assembly planning for multiple robotics arms is a critical process, and it is done for each assembly process in two steps: planning pre-assembly process and planning the assembly process itself. For the pre-assembly planning process, the process begins with the conversion of initial part configuration to the final assembly configuration (Jiménez, 2013).

For planning the assembly process, itself, the process begins with the realization of the final assembly configuration to extract the assembly sequences of each part in assembly operation (Tian et al., 2017). There is no issue regarding the second step where all assembly parts are reassembled in order according to the final assembly configuration. However, carrying out the first stage is essential since it is closely related to the planning the multiple robotics arms assembly operation in a collaborative manner that could lead to cause physical damages to the real robots during the planning stages.

However, digital representation of the production environment and process has developed as a way to handle some of the production system performance optimization (Kousi et al., 2018). The benefits that the Digital Twin technology may provide in terms of giving perception and cognitive capacities towards more autonomous and intelligent robotic systems have made it a hot topic in this era of digitalization in manufacturing (Chryssolouris et al., 2009). By running numerous simulations in the virtual world before carrying out the actual operation in the physical world, a digital twin (DT), a virtual replica of the real product or process, can be used to improve production efficiency and guarantee safety characteristics for the physical production (see in Figure 2.6).

Figure 2.6

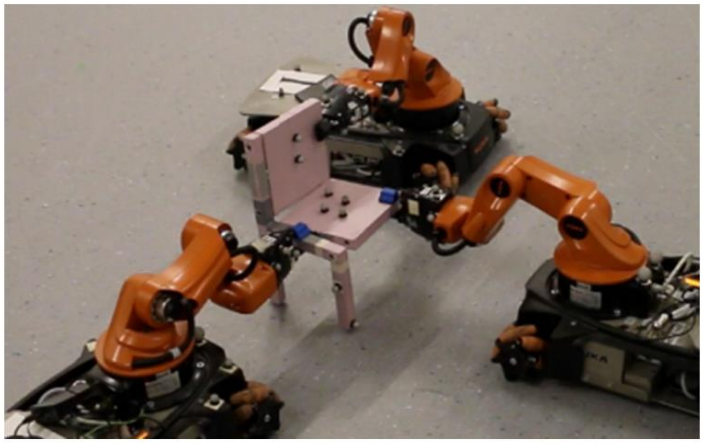
Digital Twin for Multi-Robotics Assembly Planning



Very few research has looked at this issue. In the first one, "Multiple Mobile Robots Planning for Furniture Assembly: Chair Assembly" (Dogar et al., 2019), the assembly planning sequences for multiple mobile robotics arms to construct a chair was discovered (see Figure 2.7). The approach took into account restrictions between various robotics manipulators, between the mobile robotics arms, grippers and assembly components, and between mate components and partial assembly components. During a series of collaborative assembly operations in the virtual simulation world for planning phases, the problem of identifying robot configurations to grip assembly pieces is addressed by this multi-robot assembly system.

Figure 2.7

Multiple Mobile Robots at Assembly Planning.

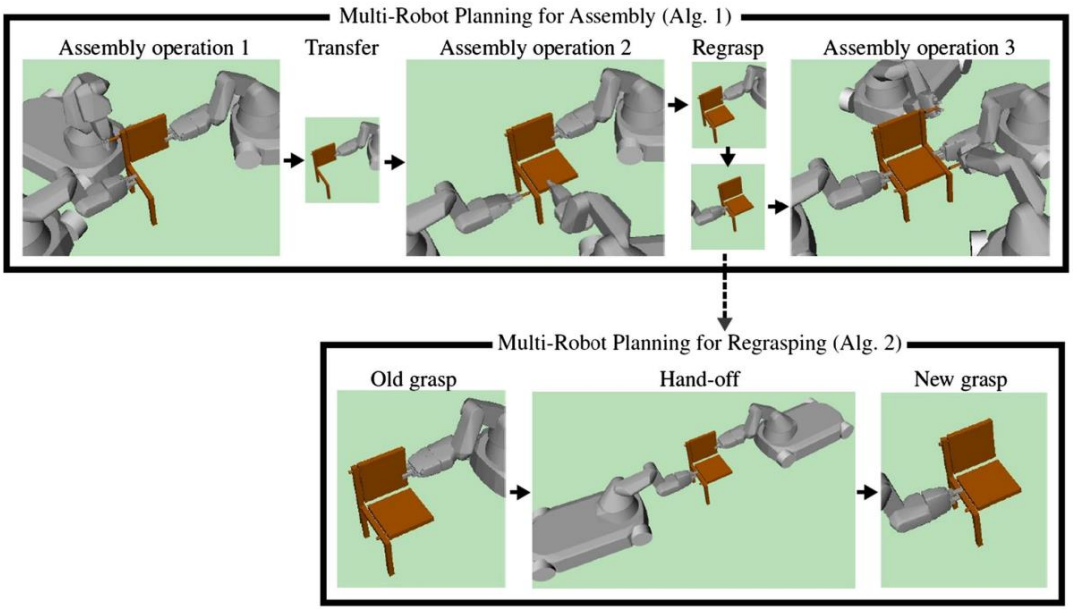


Nevertheless, the task's sequential structure may place even more restrictions. To carry a partially assembled structure from one assembly step to the next, the robots can use one of two methods (see Figure 2.8): The robots have two options: I they can regrasp the partially formed structure and change their grab, or (ii) they can transmit the partially assembled structure immediately to the subsequent operation while maintaining their hold. Both approaches have advantages.

If the robots select transfer, additional regrasp actions during execution are avoided. Regrasps, on the other hand, simplify the planning problem by disentangling the order of the operations: Since the robot in Figure 2.8 agrees to move the structure between Assembly Operations 1 and 2, it must plan a part-grasp that is suitable for both operations. Long sequences of processes are highly expensive to solve because of the connection between many procedures. In this research, simple rectangular and square shape component are used for chair assembly. Moreover, the initial part placement has not been considered yet.

Figure 2.8

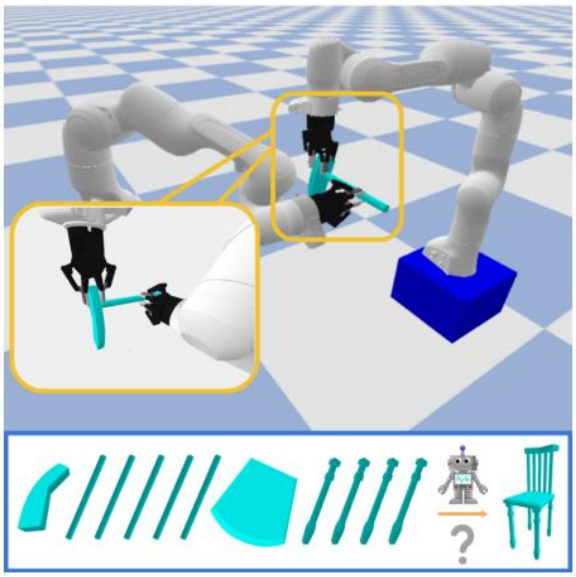
Multiple Mobile Robots at Assembly Configuration.



In a different study titled "Multi robot assembly planning system, Chair Assembly" (Yu et.al, 2021), the authors construct a robotics assembly simulation environment in which two robots must sequentially pick up two parts, plan collision-free motion trajectories, precisely mate two parts, and then return to the previous pickup movement (see Figure 2.9). The 220 chairs from the PartNet dataset (Li et al., 2019) were utilized in this investigation. The relative postures of the parts were altered to satisfy collision and contact criteria, and connection points and graspable regions were annotated. In this research, some polygon shape components are start introduced for chair assembly but the shape of the parts are still easy to find the grasp location on it. The initial placement of the assembly part is had not been discussed yet.

Figure 2.9

Multiple Robots Assembly Planning System: Chair Assembly (Yu et.al, 2021).

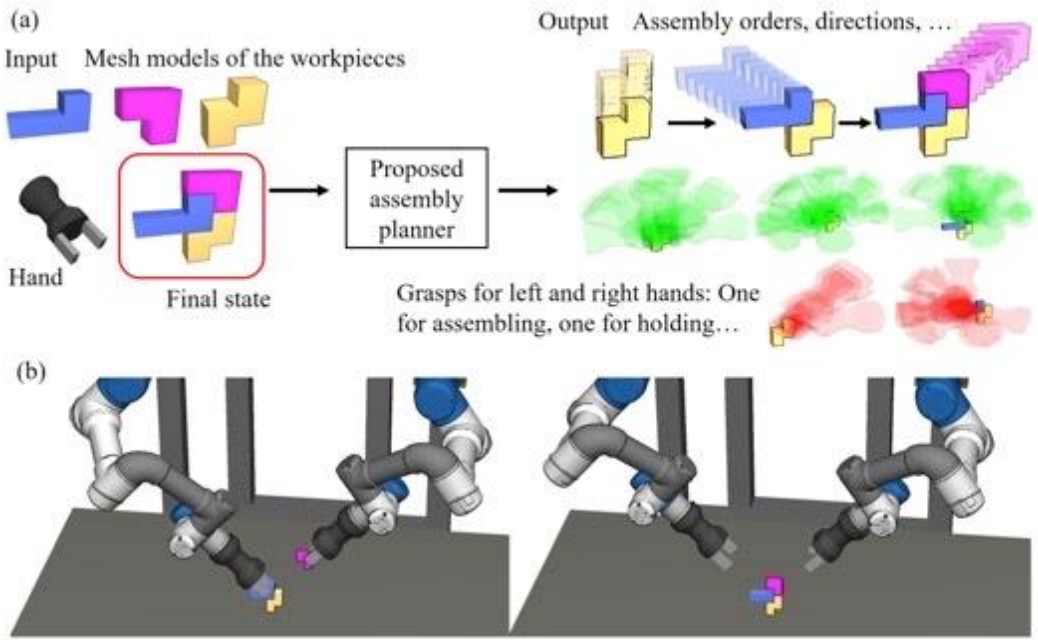


The next article, "Dual Arm Assembly Planning for Soma Block Assembly," by (Chen et al., 2021), describes a planning strategy that can spontaneously determine the optimal assembly sequences for a dual-arm robotics manipulator to assemble the soma blocks (see Figure 2.10). The used planning method generates all potential assembly sequences using the 3D-mesh model of the objects and the final assembled state, and then evaluates the best assembly sequences by taking graspability, assemblability, stability, and the requirement for a second robotics arm into account.

The system presents a way that uses a second robotics arm to handle the unstable 3D components on the work table, which is the key improvement. It also develops a more reliable stability analyzer. Before running the physical system, the planner is tested in the virtual simulation environment. The researchers used simple shape soma block for assembly process and the initial placement of the part are not discussed in this research.

Figure 2.10

Dual Arm Assembly Planning for Soma Block Assembly.



According to previous research, the researchers still hadn't considered the handling of complex shape 3D parts and still haven't touched the research area on random part placement in initial configuration. Handling the complex shape 3D parts which grasp's locations are difficult to find that are placed in initial random configuration becomes the challenging research area to conduct in multi-robot assembly system.

CHAPTER 3
CONCEPT GENERATION

3.1 Idea Generation

The 3D robotics assembly system arranges the assembly sequence step by step, according to information from a final assembly configuration. As aforementioned, the robotics assembly planning is a critical process, and it is done for each assembly process in two steps: planning pre-assembly process and planning the assembly process itself. There is no issue regarding the second step where all assembly parts are reassembled in order according to the final assembly configuration.

However, employing the first stage is crucial since it closely relates to the complexity of the assembly system when it starts to consider the handling of random configuration of complex 3D parts in initial setup. Furthermore, the problem becomes more complex when the used assembly parts are in complex shape with random orientation, and they must be converted to the final assembly configuration from any random initial configuration.

Some researchers conducted this area with the use of simple 3D objects for instance, soma blocks, Lego blocks, rectangular shape or square shape part which are still far to reach the real industrial assembly applications. Therefore, the main idea of this study becomes to reach real life industrial assembly applications by handling complex shape 3D parts in random initial configuration setup. Developing the multi-robot assembly system for complex 3D parts in random initial configuration in the physical world can create not only time consuming but also physical damages to the real robots.

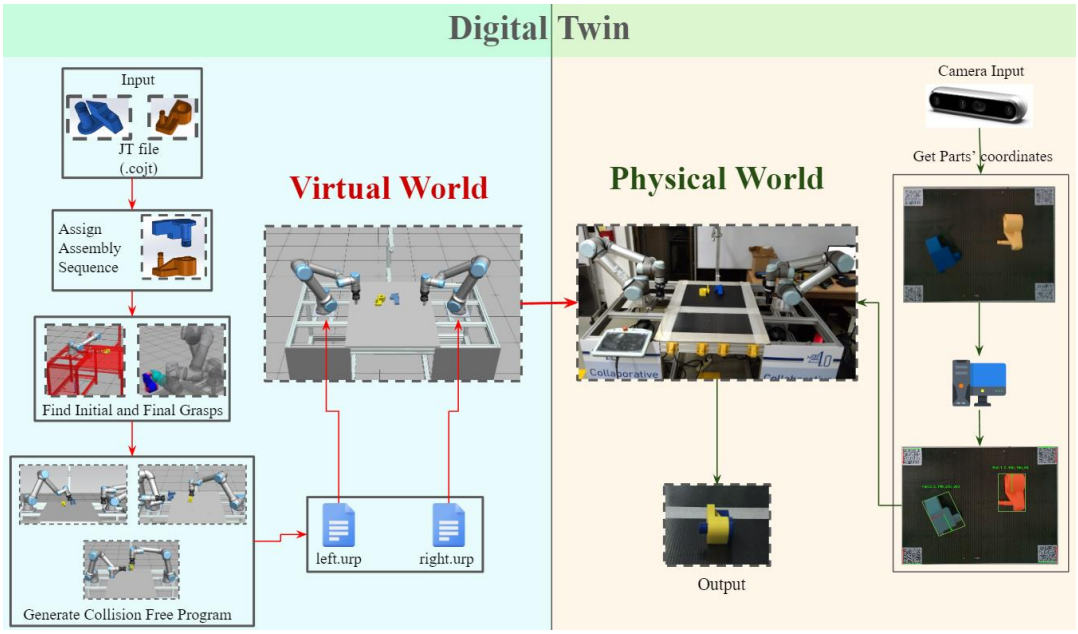
Therefore, the digital twins become an important technology in order to conduct multiple simulations in the virtual world to ensure whether there are collisions or not in the pre-assembly planning process in Co-MRAS. This research aims to develop the digital twin system for 3D assembly process of Co-MRAS to better support the handling of complex 3D parts in random configurations at the initial state.

The digital twin system for both physical world and virtual world is illustrated in Figure 3.1. The virtual world is implemented to conduct mainly for pre-assembly planning process such as verification of initial and final grasp’s location on each part and conduct

multiple simulation for generating different robotics program for multiple robotics arms. In the physical world, the physical robots collect the digital twin robotics program from virtual world and the camera take the snapshot and analyze the coordinates data to transfer to the physical robots. The new parts' coordinates will be replaced in the predefined digital twin robotics program.

Figure 3.1

Digital Twin for Co-MRAS

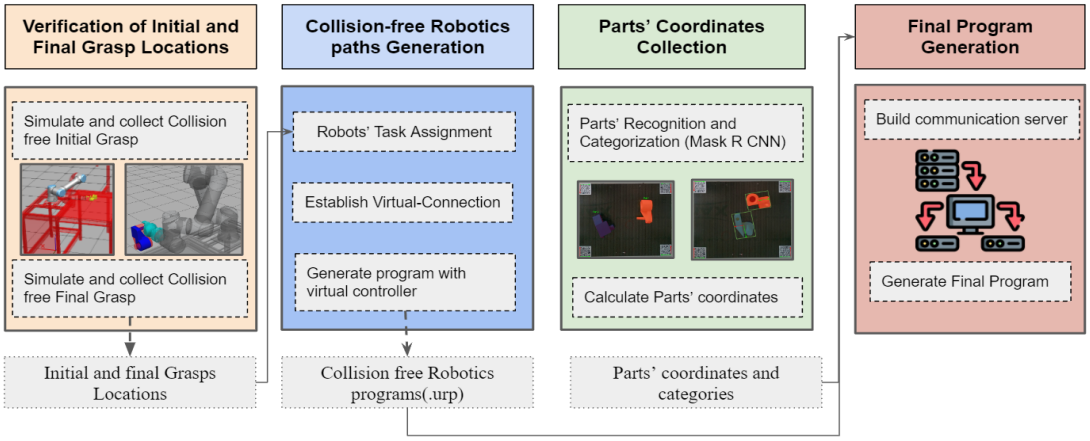


3.2 Collaborative Multi-Robotics Assembly System (Co-MRAS)

Collaborative multi-robotics assembly system (Co-MRAS) can be defined as multiple robotics arms working together in a shared workspace to acquire the sub-assembly component or final assembly product. But dealing with complex 3D parts in random initial configuration for Co-MRAS create four main challenges: (1) extracting exact initial grasp location to convert to final assembly configuration, (2) allocating the common task on multiple robots in shared workspace, (3) recognizing the different random initial configuration of parts in 3D space and last but not least, (4) generating collision free motion for multiple robotics arms in a shared workspace. There are four primary phases to take in order to complete this task while taking the difficulties into consideration, and Figure 3.2 provides a summary of the process planning for Co-MRAS.

Figure 3.2

Co-MRAS Process Planning Overview



The first step focuses on the geometric constraints of the complex-shape objects in 3D space and their stable placement on the work table to identify the collision free grasp's location on each part in every initial random configuration. The second step focuses on the assembly motion planning for multiple robotics arms according to the collision free initial and final grasp's location by considering the coordination of multiple robots and collision among them. The third part focuses on the collection of parts' coordinates from the real world in order to detect the position and rotation information of each work piece and last but not least the fourth step focuses on the final program generation to the physical robots.

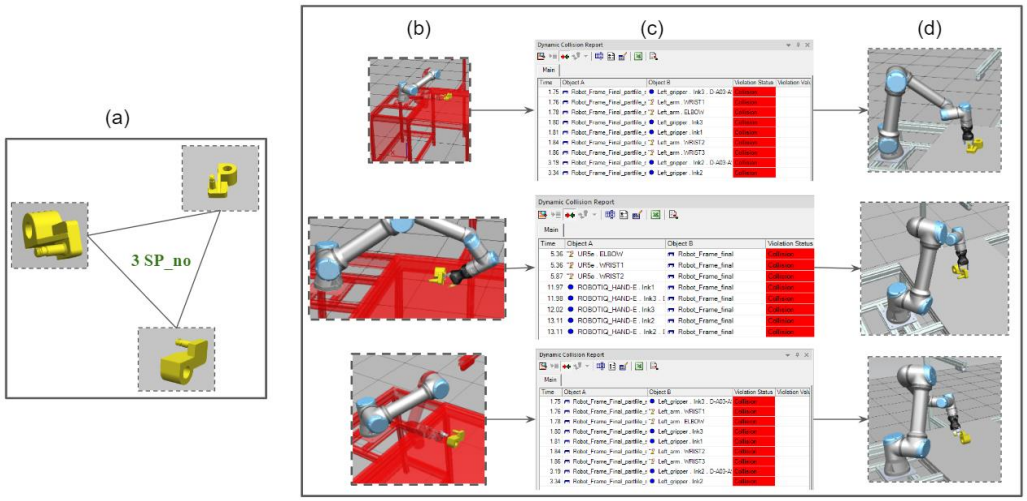
3.2.1 Verification of Initial and Final Grasps Location

As aforementioned, the main objective of this step is to generate the collision free initial and final grasp's locations according to the number of stable placements of each work piece on the work table. The inputs for this step are information about all possible stable placement numbers of each work piece on the work table according to their center of gravity to verify the collision free grasps by conducting different multiple simulations. According to the different stable positions of each work piece, the operator needs to identify and confirm the initial and final grasp's location by doing simulations in the virtual world. In the virtual simulation environment, object-oriented simulation is conducted by giving the geometric constraints: (1) center of gravity, (2) parallel and cylindrical faces, and (3) final assembly pose for each assembly component.

***3.2.1.1* Initial Grasps Location Verification.** In order to identify the collision free grasp for the initial state, the first step is that the system needs to recognize the stable placement no of each part on the work table by using the geometric constraint of center of gravity of each work piece. Each component can have multiple stable placement numbers according to their center of gravity. (See Figure 3.3 (a)). After once knowing the stable placement of each part, the operator needs to identify the collision free grasp between the robotics gripper and work table by conducting simulation in the virtual world as mentioned in Figure 3.3 (b). The simulation results will show all the collision grasps as illustrated in Figure 3.3 (c). According to the collision checking result, the operator needs to remove all collision grasps and verify the collision free grasps as shown in Figure 3.3 (d).

Figure 3.3

Verification of Initial Grasp Location



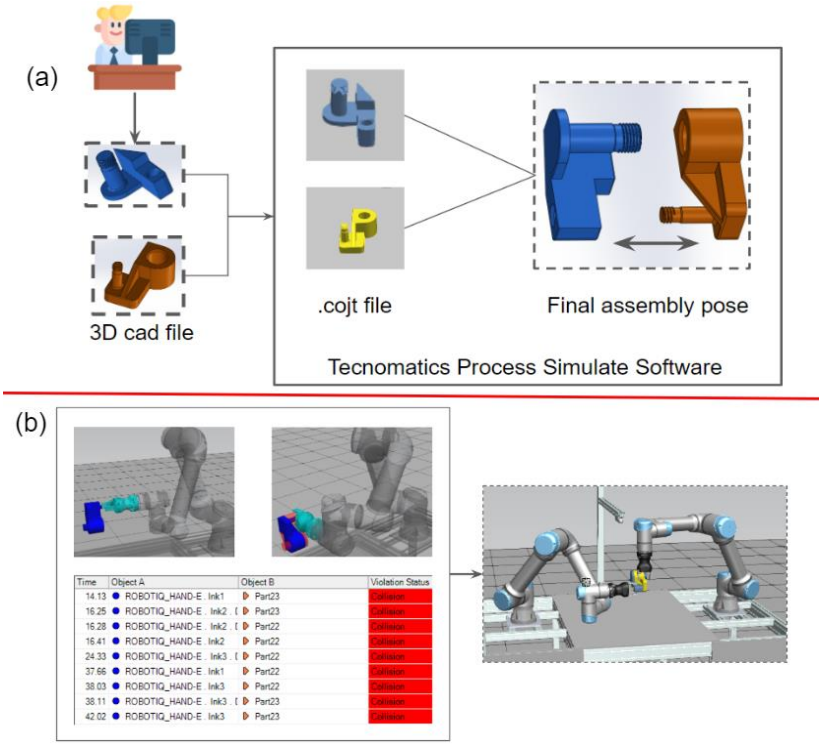
Notes: (a) Stable Placements of Part 1; (b) simulate defined grasps; (c) collisions grasps results (d) Collision free initial grasp's locations

***3.2.1.2* Final Grasps Location Verification.** In this section, the operator needs to find the collision free final grasp location of each component. The first step is done by assigning the final assembly pose for every assembly component as shown in Figure 3.4 (a). The inputs of this method section are 3D (.cojt) files for each assembly component and their final assembly poses. Then, according to that final assembly pose, the collision checking simulation between the gripper and partially assembled components is conducted to identify all collision grasps at the final state of the assembly

process as shown in Figure 3.4. All the collision grasps are removed to get the collision free final grasps location for each robotic arm. So, the output is collision free final grasp's location of each assembly work piece for multiple robotics arms as you can see in Figure 3.4 (b).

Figure 3.4

Verification of Final Grasp Location



Notes: (a) Assign Final Assembly Pose; (b) Collision Free Final Grasp Location

3.2.2 Collision Free Robotics Path Generation

After the system gets the initial and final grasps location of each assembly component, this section presents the method that can generate the collision free robotics paths according to the different initial part's placement on the worktable that has been presented in section 3.1.

***3.2.2.1* Robots' Task Assignment.** The number of different non repetitive tasks can be generated by performing a simulation of stable placement on the work table in the virtual world according to the number of assembly parts and their different stable placement. In figure 3.5 (a), the system has two assembly parts to perform the assembly process and each part has different initial stable placement. According to those different

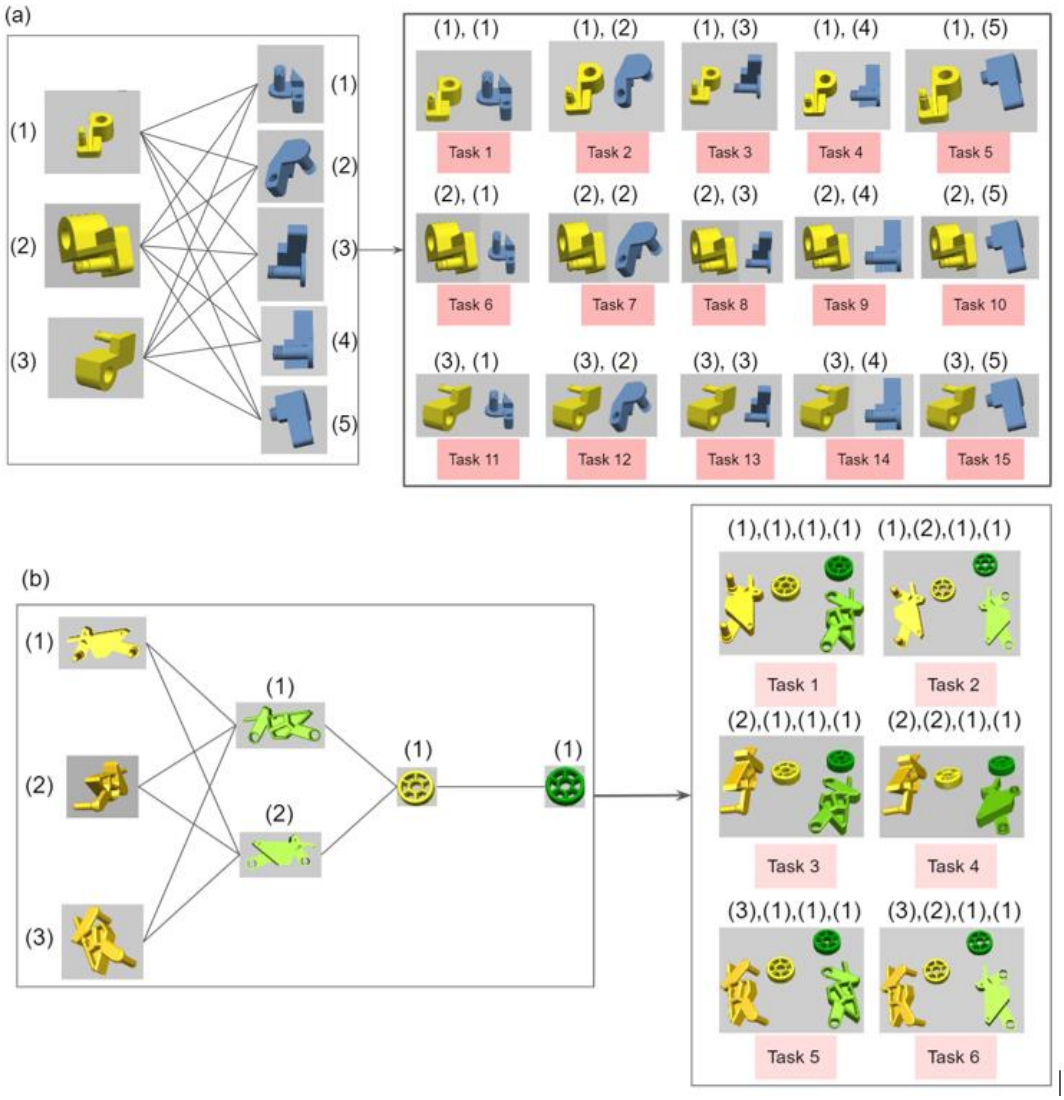
initial stable placements among two parts, the system will have different non repetitive tasks for each robot to perform the assembly process. The system used equation 1, in order to generate the different initial placement. In equation 1, total number of initial stable placement is denoted as Total SP, number of stable placements in Part 1 is denoted as SP_1 and the number of stable placements in Part n is denoted as SP_n. The generated all possible different non repetitive tasks are shown in Figure 4.

$$Total\ SP = SP_1 \times SP_2 \times \cdots \times SP_n \qquad \text{eq. (1)}$$

In Figure 3.5 (a), the yellow part has 3 different stable placements and the blue part has 5 different stable placements on the work table. So, the system generates 15 different task assignments to multiple robots. In Figure 3.5 (b), the yellow cycle body has 3 different stable placements, the green cycle body has 2 different and the other two wheels have 1 stable placement on the work table. So, the system generates 6 different robotics tasks for multiple robotics arms.

Figure 3.5

Different Non-Repetitive Tasks According to Initial Stable Placements

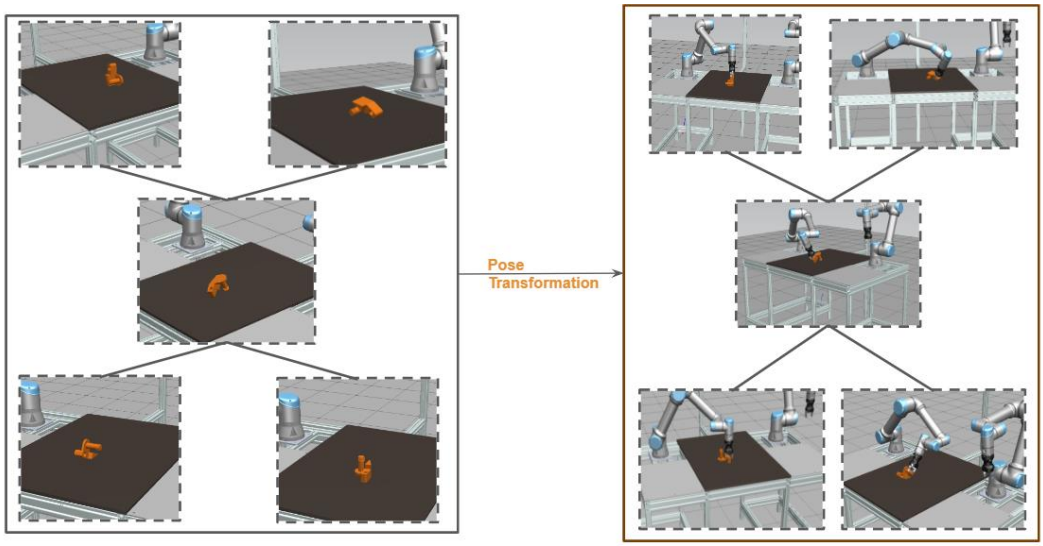


Notes: (a) Bracket Assembly Case (15 different non repetitive tasks); (b) Cycle Assembly Case (6 different non repetitive tasks)

***3.2.2.2* Robotics Path Generation to Initial Grasp Location.** Since the system already has Initial and final grasp's locations for each stable placement from the previous step, the system can generate the different robotics paths to initial position of the assembly object in the digital twin platform as shown in Figure 3.6. So, the path will change according to the initial placement of each part.

Figure 3.6

Collision Free Robotics Paths to Different Initial Stable Placements



In order to reach the initial grasps location from the reference pose of the robot, equation (2) the position vector and the eq. (3) rotation vector are used to transform the pose of the robot. The rotation vector is needed to convert by using *rotvec2rpy (rotation vector)* command in the Polyscope programming platform to get rotation matrix *R*. After the system once got the position vector *P* and rotation matrix *R*, the equation (4) pose transformation matrix can be generated in order to reach the initial grasps locations from the robot's reference pose automatically in the virtual world.

$$ {}^{A}_{0}P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \qquad \text{eq. (2)} $$

$$ Rotation\ Vector = \begin{bmatrix} \theta_{\mu_x} \\ \theta_{\mu_y} \\ \theta_{\mu_z} \end{bmatrix} \qquad \text{eq. (3)} $$

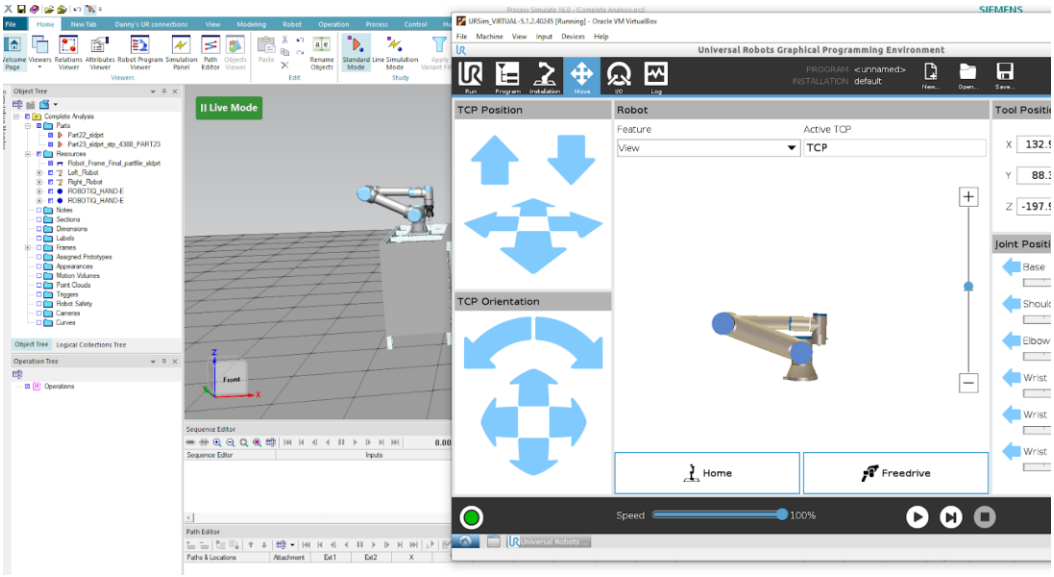
$$ T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \qquad \text{eq. (4)} $$

***3.2.2.3* Generate Collision Free Path with Virtual Controller.** In order to develop collision free predefined paths for different non repetitive tasks in the virtual

world as mentioned in Figure 3.4, the first thing the system needs to do is establish a live connection between the virtual Polyscope controller and digital twin server to generate the (.urp) program which can run in the robotics controller directly. The collision free paths can be generated by assigning predefined paths for multiple robotics arms according to the assembly sequence. Figure 3.7 represents the status of live connection between virtual Polyscope controller and Tecnomatix process simulate software.

Figure 3.7

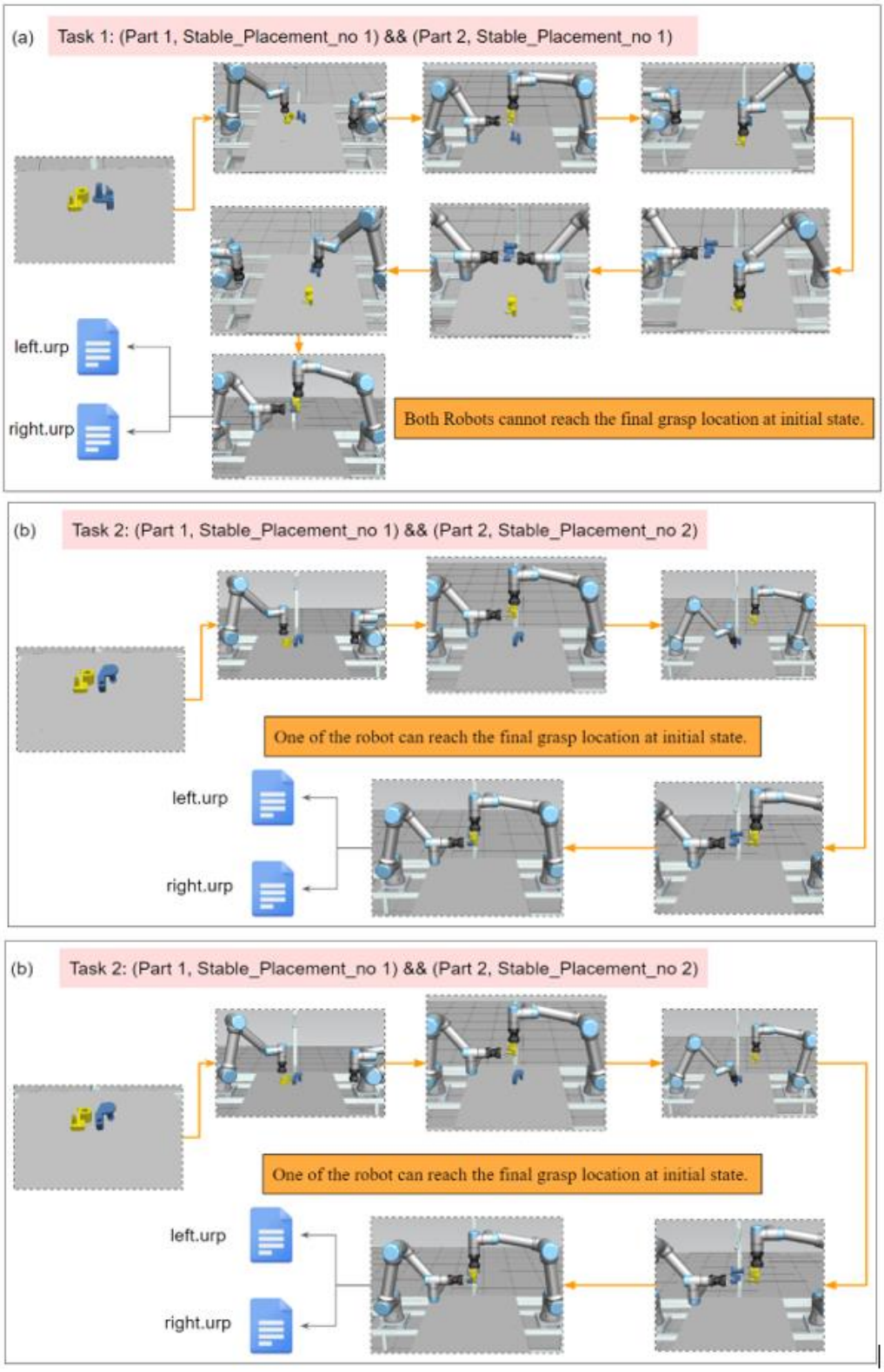
Live Connection Status Between Virtual Controller and Digital Twin Platform



Once the system establishes the live connection between two platforms, the predefined paths can be generated according to the different non repetitive tasks for both robotics arms in digital twin software. After that, the virtual Polyscope controller recorded the all-predefined way points that the system generated in the Tecnomatix platform. In Figure 3.8, The three different predefined non repetitive tasks are illustrated in Figure 3.8 (a), (b) and (c).

Figure 3.8

Different Predefined Robotics Paths in Tecnomatix Process Simulate Software

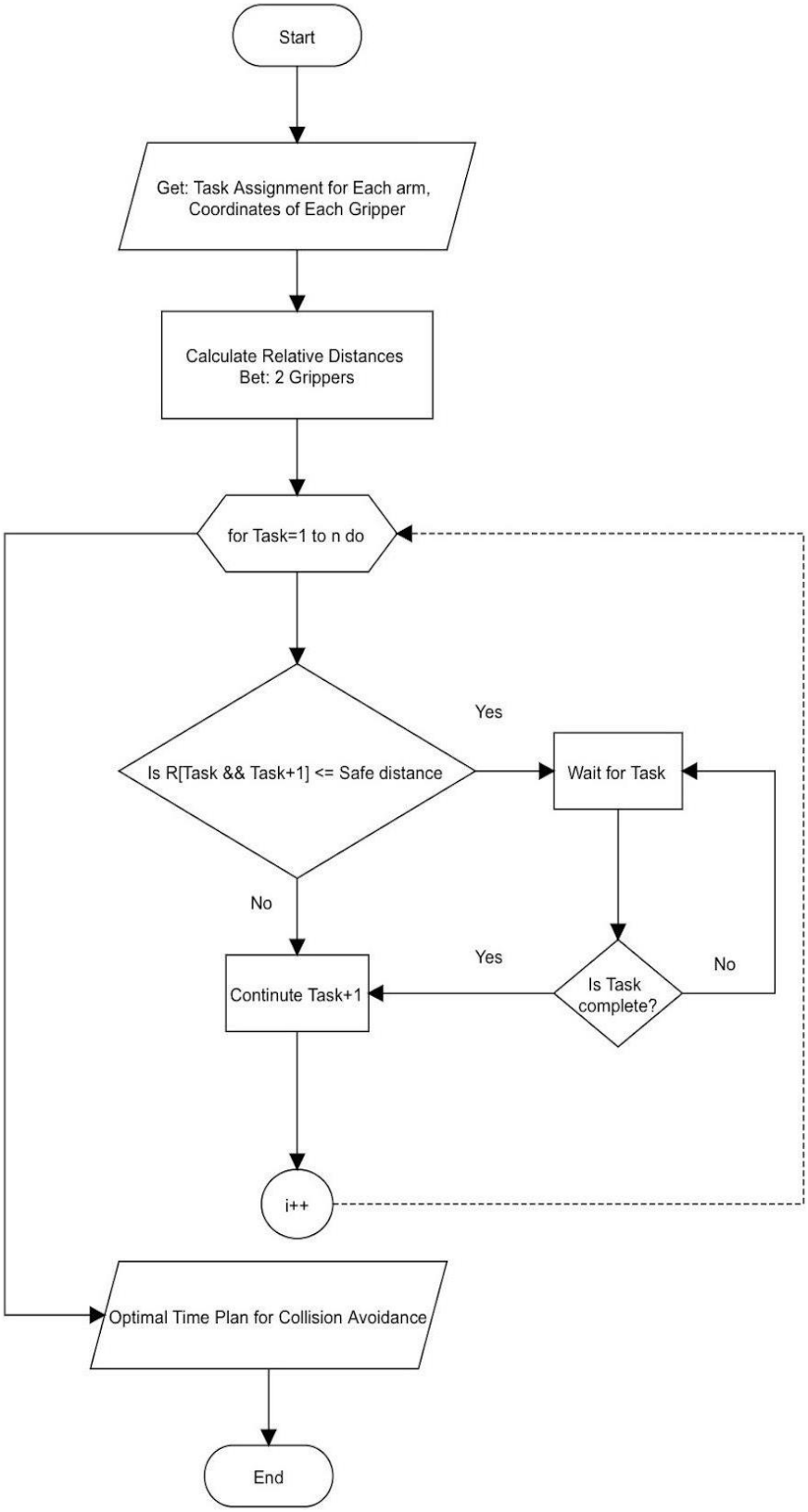


Notes: (a) Task 1: (Part1, SP1) & (Part2, SP1); (b) Task 2: (Part1, SP1) & (Part2, SP2); (c) Task 7: (Part1, SP2) & (Part2, SP2)

All developed robotics paths are simulated in the digital twin platform in order to check collision among multiple robotics arms. But there are still chances to have collisions while the multiple robots are in part picking positions. For avoiding the collision among multiple units at the part picking stage, the suggested algorithm is used to remove the collision paths by using set and wait commands among multiple robotics arms to avoid collision. Figure 3.9 shows the algorithmic flowchart for collision avoidance among multiple robotics arms.

Figure 3.9

Collision Avoidance Algorithm Among Multiple Robotics Arm



3.2.3 Parts' Coordinates Collection

This section describes the parts' recognition, categorization and part's coordinates collection from the real-world coordinate system in order to acquire the position and rotation information of each assembly component. There are two main sections in Parts' Coordinate Collection method. They are; Part's detection and categorization section, and Calculate Parts' coordinates section.

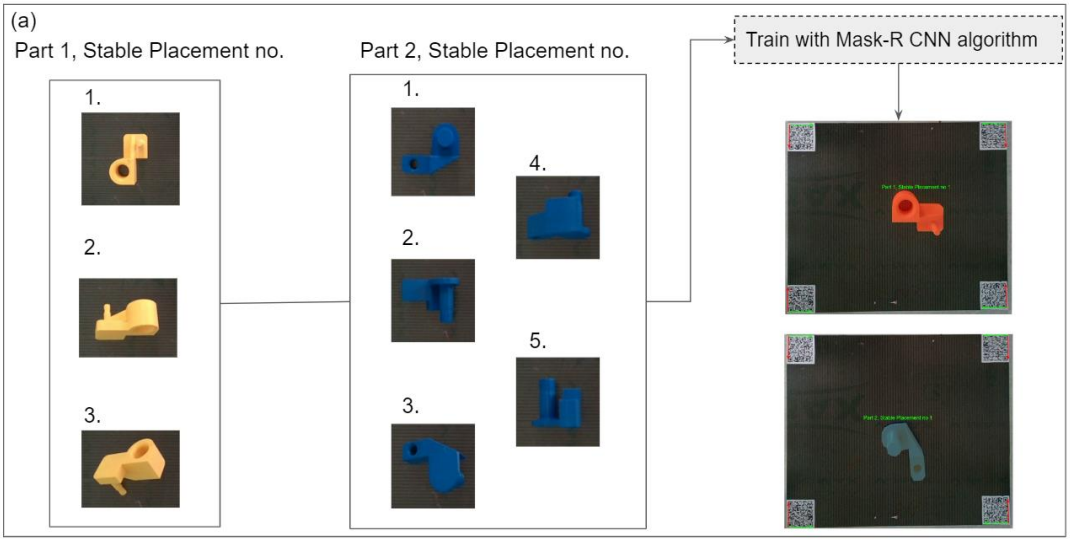
***3.2.3.1* Parts' Detection Module.** For Parts' detection, Mask-R CNN algorithm is used to train different parts to recognize by the system. The system's Training Module, which seeks to produce a useful dataset, is where it all begins. A typical RGB camera is used to gather the training photos, which are then shrunk and tagged. The labeled image and the .json files produced throughout the process are used to begin the training process after that. The trained model is eventually produced and used in the detection module.

The picture inputs, or images of the objects' XY direction (from the top) and Z direction, serve as the starting point for the detection procedure (from the side). Both photos go through a pre-processing step to adjust brightness, contrast, and other aspects using techniques like High Dynamic Range or denoise. Before sending the image to the detecting module, the pre-processing technique can also resize, rotate, and mirror it as necessary.

On the basis of the model developed in the Training Module, the Detection technique then begins to identify the items in the photos. The outcome is a binary mask together with the object's detected class. For the training module, the proposed method categorizes the images according to the part and its stable placement number as shown in Figure 3.10 (a). 500 training images are used for each part to get the accurate result. The output of the training are the parts' that can be recognized by the system with masking.

Figure 3.10

Categorization Training Images for Recognition

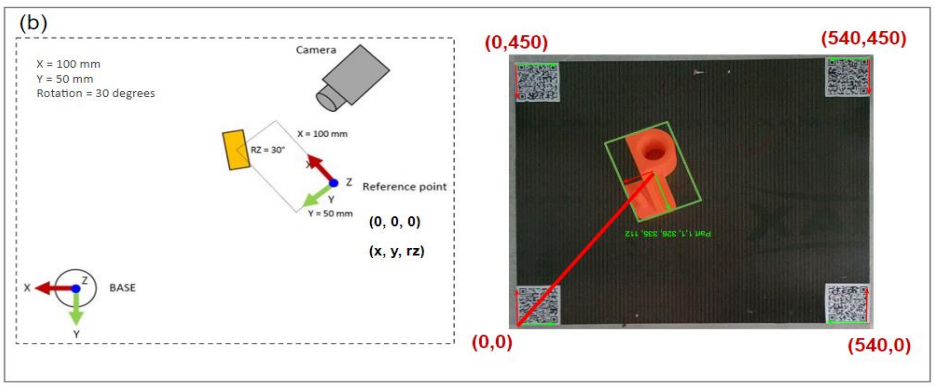


***3.2.3.2.* Parts' Coordinate Calculation.** After Parts' recognition is done, to calculate part's coordinates, the fixed camera frame method is set up as shown in Figure 3.11. If the part is placed in the camera's field of view, the camera recognizes the part, relative to the reference point, is denoted as offset. As the real-world reference points, the 4 QR codes are used to calculate the target by using translation and rotation matrix from the reference point, as presented in equation 5.

$$Target = pose\ trans\ (reference\ point, offset) \qquad \text{eq. (5)}$$

Figure 3.11

Fixed Camera Position and Same Reference Point Setup



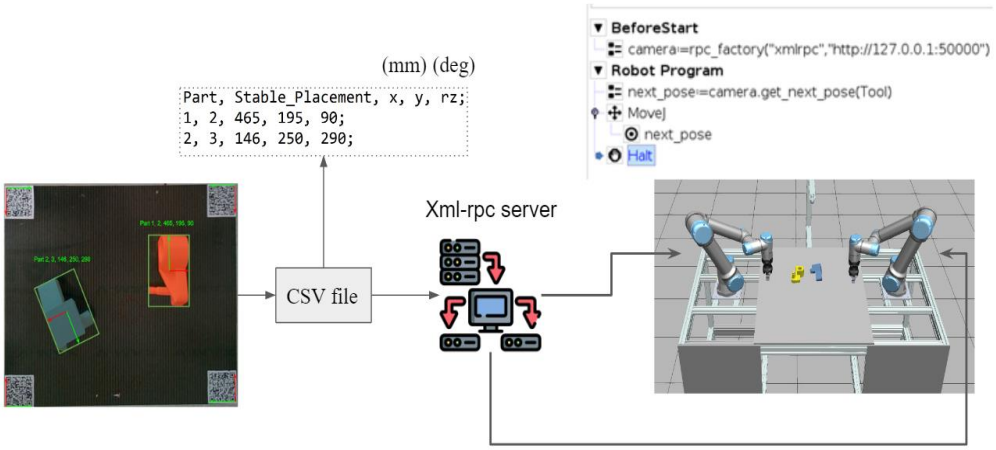
3.2.4 Final Program Generation

After the system gets the collision free robotics paths and collects part's coordinates, the proposed system is needed to transfer the real time parts' coordinates from the camera to multiple robotics arms. In this section, the communication method will be discussed in order to transfer the real time parts' coordinates data to the physical robotics arms. After the system gets the two .urp files for both robotics arms, those final programs are needed to transfer to the physical robots from the virtual controller to the physical one.

The new coordinates of parts are needed to send to a predefined robotics program. In order to transfer the coordinates, the proposed method is needed to develop the connection between PC and robots by using XML_RPC server. From the image calculation section, the output of the Parts' coordinate is collected as a CSV file. The CSV file is uploaded to the XML_RPC server to send the file to the robot. In the robotics controller, the program call is needed to run the XML_RPC server Before Start Sequence as shown in Figure 3.12.

Figure 3.12

Image Coordinates Transfer to Robots with XML-RPC Server



CHAPTER 4
SYSTEM DEVELOPMENT

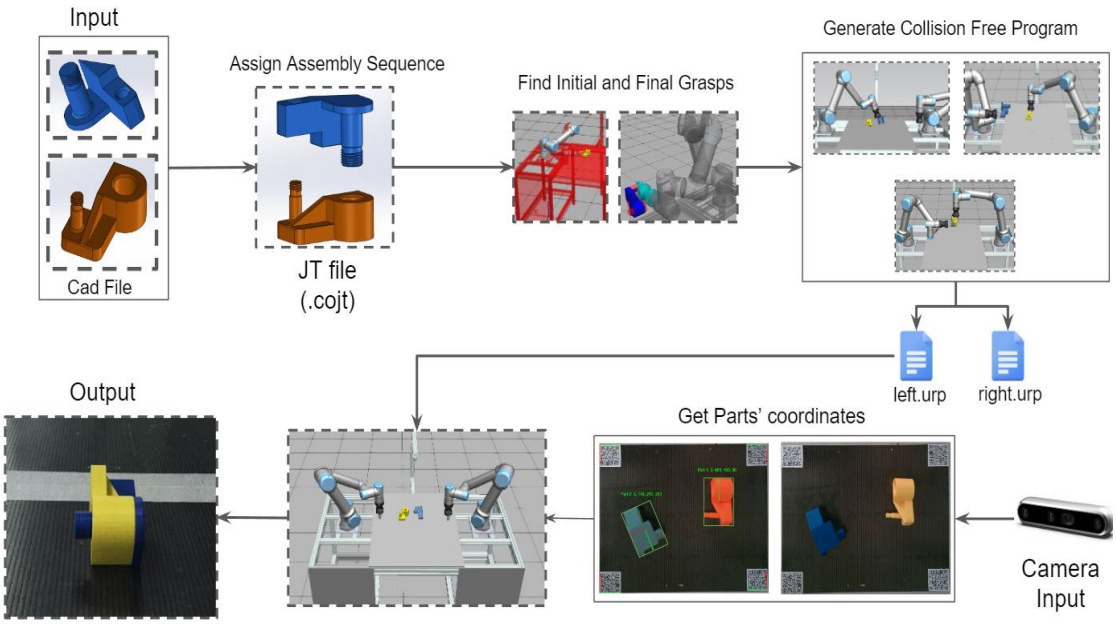
The process of transforming the developed concept into an algorithm and putting it into use with two robotic arms is explained in this chapter. This section provides a detailed explanation of the autonomous Co-MRAS development process, including the creation of algorithms, the creation of digital twins, and the creation of the coordinate collection application.

4.1 Algorithm Development

The Autonomous Co-MRAS primarily emphases on the virtual simulation which include collision identification for collision avoidance and image processing which include part's recognition and coordinate collection. The created idea is then turned into an algorithm to finish the assembly process beginning with the CAD file in order to put this into practice. The created algorithm's flow diagram is shown in Figure 4.1.

Figure 4.1

Developed Algorithm for Autonomous Co-MRAS



4.2 Digital Twin Development

In order to accomplish the proposed algorithm, the digital twin program is built to find the collision free robotics paths for each robotic arm. The Tecnomatix Process Simulate (TPS) software is used to conduct multiple simulations to extract the collision free initial and final grasp's location and develop the different non repetitive tasks for each robot according to the different initial random placement of each assembly component.

4.2.1 Tecnomatics Process Simulate (TPS): Digital Twin Software

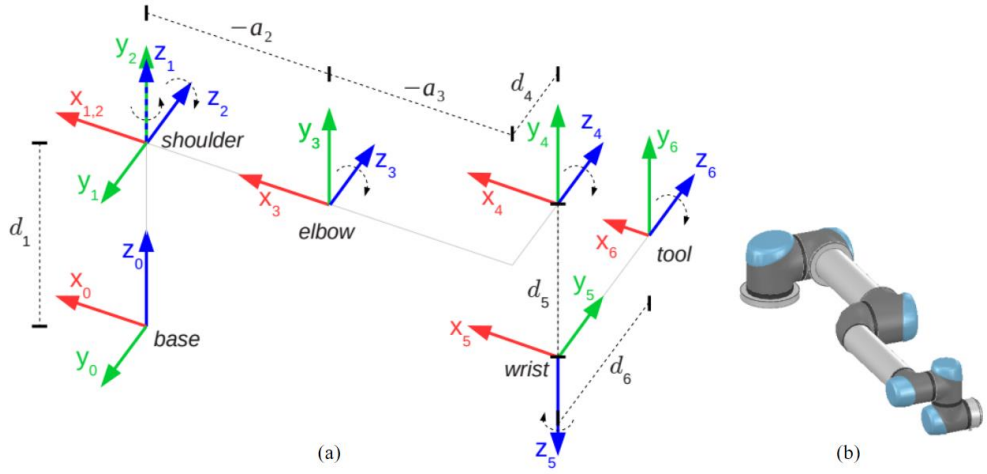
A group of engineering research tools marketed under the name Process Simulate are available from Process Simulate Human, Process Simulate Robotics, Process Simulate Assembler (flow components), and other features are included in this suite. Process Simulate is a dynamic environment that makes it simple for a researcher to do assembly and serviceability studies as well as concept verification.

4.2.1.1 **Robotics Arm Modelling Forward Kinematics (FK).** In TPS digital twin platform, the virtual layout of the physical objects can be set up, visualize and operate them as a virtual world. All the virtual 3D models that are created as same as physical world can able not only to modify kinematics pairs of 3D objects for tooling and robots but also allow to check collisions and clearances before collision can be calculated dynamically between simulated models.

In modelling of kinematics chains, there are some basic functions that need to be used in the process of modeling. A group of things within a component that remain fixed, relative to one another, is referred to as a rigid body or The Link. The joint is a kinematic chain's fundamental moving component. These letters are the first letters of the default names in TPS software: LNK (for example LNK1, LNK2, LNK3). An axis, two links, and a joint make up a joint. It describes how these two links move in relation to the axis. Figure 4.2 depicts the UR5 robot's kinematic framework at the initial position.

Figure 4.2

UR5e Robot in Zero Position



Notes: (a) Description Frames for the Robot UR5e; (b) 3D Visualization Model

The forward kinematics and inverse kinematic equations of a UR5e robot can be derived in accordance with the reference on kinematics modeling. Inspired by Hawkins (2013), Keating (2017), and Kebria et al. (2016), the reference of kinematics equations aims to explain each phase in greater detail. The reference equation uses the Denavit-Hartenberg nomenclature, often known as modified DH-parameters, as it was used by Craig (2005). Additionally, the use of the succinct notations ${}^{0}_{6}P$ is the frame 6 original as seen in frame 0, and ${}^{0}_{6}\hat{Y}$ is a unit vector that indicates how the y-axis of frame 6 appears when seen from frame 0. ${}^{0}_{6}T$ is a Transformation matrix from frame 6 to 0.

$$ {}^{0}_{6}P = \begin{bmatrix} {}^{0}_{6x}P \\ {}^{0}_{6y}P \\ {}^{0}_{6z}P \end{bmatrix} \; {}^{0}_{6}\hat{Y} = \begin{bmatrix} {}^{0}_{6x}\hat{Y} \\ {}^{0}_{6y}\hat{Y} \\ {}^{0}_{6z}\hat{Y} \end{bmatrix} \qquad \text{eq. (6)} $$

$$ P^{0} = {}^{0}_{6}T \,.\, P^{6} \qquad \text{eq. (7)} $$

Based on known joint θ_{1-6}., the forward kinematic (FK) equations compute a transformation matrix ${}^{0}_{6}T$. (1-6). The definition of the transformation matrix is:

$$ {}^{0}_{6}T = \begin{bmatrix} {}^{0}_{6}R & {}^{0}_{6}P \\ 0 & 1 \end{bmatrix} \qquad \text{eq. (8)} $$

$$= \begin{bmatrix} {}^{0}_{6}\hat{X} & {}^{0}_{6}\hat{Y} & {}^{0}_{6}\hat{Z} & {}^{0}_{6}P \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{eq. (9)}$$

$$= \begin{bmatrix} {}^{0}_{6x}\hat{X} & {}^{0}_{6x}\hat{Y} & {}^{0}_{6x}\hat{Z} & {}^{0}_{6x}P \\ {}^{0}_{6y}\hat{X} & {}^{0}_{6y}\hat{Y} & {}^{0}_{6y}\hat{Z} & {}^{0}_{6y}P \\ {}^{0}_{6z}\hat{X} & {}^{0}_{6z}\hat{Y} & {}^{0}_{6z}\hat{Z} & {}^{0}_{6z}P \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad \text{eq. (10)}$$

where the axes of frame 6 with reference to frame 0 are defined by the columns ${}^{0}_{6}\hat{X}$, ${}^{0}_{6}\hat{Y}$, and ${}^{0}_{6}\hat{Z}$. As a series of transformations, one for each joint, the transformation matrix can be divided as follows:

$${}^{0}_{6}T(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = {}^{0}_{1}T(\theta_1){}^{1}_{2}T(\theta_2){}^{2}_{3}T(\theta_3){}^{3}_{4}T(\theta_4){}^{4}_{5}T(\theta_5){}^{5}_{6}T(\theta_6) \qquad \text{eq. (11)}$$

For the UR5e 6DOF robotics arm manipulator, the DH parameters are specified as the Table 4.1 below.

Table 4.1

Modified D-H Parameters Table for a UR5e Robot

i	$\alpha_i - 1$	$a_i - 1$	d_i	θ_i
1	0	0	d_1	θ_1
2	$\alpha_1 = 90^{\circ}$	0	0	θ_2
3	0	a_2	0	θ_3
4	0	a_3	d_4	θ_4
5	$\alpha_4 = 90^{\circ}$	0	d_5	θ_5
6	$\alpha_5 = -90^{\circ}$	0	d_6	θ_6

The equation can be derived for the transformations for each connection using the DH-parameters. Given are the general transformations between link *i* -1 and *i*:

$${}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_i - 1 \\ \sin\theta_i \cos(\alpha_i - 1) & \cos\theta_i \cos(\alpha_i - 1) & -\sin(\alpha_i - 1) & -\sin(\alpha_i - 1)d_i \\ \sin\theta_i \sin(\alpha_i - 1) & cos\theta_i \sin(\alpha_i - 1) & \cos(\alpha_i - 1) & \cos(\alpha_i - 1)d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{eq. (12)}$$

It is forthright to described DH-parameters from Table 4.1 in equation (12) to create the transformation matrices for each link of the UR5e robot. Then, using equation (11), the

total transformation from base to end-effector can be obtained by multiplying all six transformation matrices. The transformation matrix ${}^{0}_{6}T$'s 12 parameters have analytical equations as a result. It can be found all of the analytical equations in (Hawkins, 2013).

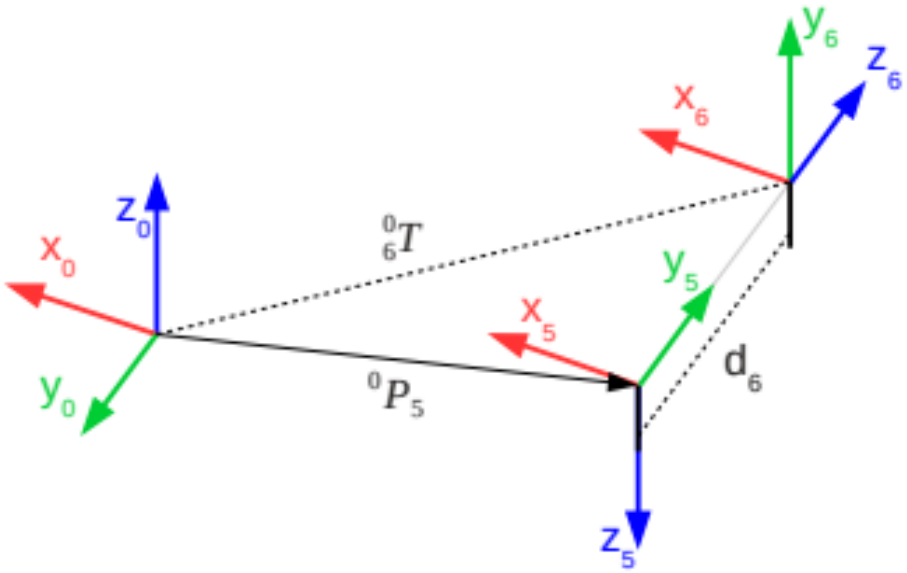
It is simple to enter the DH-parameters from Table 4.1 in equation (12) to create the transformation matrices for link of the UR5e robot separately. Then, using equation (11), the total transformation from base to end-effector can be obtained by multiplying all six transformation matrices. The transformation matrix ${}^{0}_{6}T$, the 12 parameters have analytical equations as a result. The derived equation can be found in all of the analytical equations in (Hawkins, 2013).

4.2.1.2 **Robotic Arm Modelling Inverse Kinematics (IK).** The joint angles θ_{1-6} are calculated using the inverse kinematic (IK) equations based on the end frame's desired position and orientation, which is defined as the transformation ${}^{0}_{6}T$. This research limits all angles in the answer to $(\theta_1, \ldots\ldots, \theta_6)\ \epsilon\ [0; 2\pi]$.

This research first establishes where the wrist frame 5 lies in reference to the base frame; ${}^{0}_{5}P$, in order to find θ_1. It is possible to find ${}^{0}_{5}P$ by going backwards from frame 6 to 5, as shown in Figure 4.3. Keep in mind that d_6 and ${}^{0}_{6}T$ are both known.

Figure 4.3

Locating the Fifth Frame's Origin



The translation ${}^0_5P$ can be described as in equation 12.

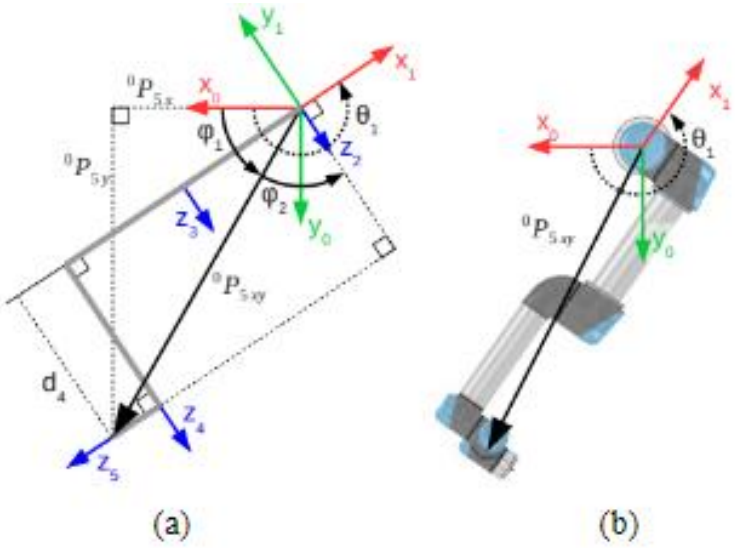
$$ {}^0_5P = {}^0_6P - d_6 \cdot {}^0_6\hat{Z} $$

$$ {}^0_5P = {}^0_6T \begin{bmatrix} 0 \\ 0 \\ -d_6 \\ 0 \end{bmatrix} \qquad \text{eq. (13)} $$

In order to arrive at θ_1, we look at the robot in Figure 4.4's Top view (looking into z_0):

Figure 4.4

Robot (Until Frame 5) as Viewed from Top View



Notes: (a)Frames Describing with Vector the UR5e Robot; (b) 3D Top View Visualization Model

The equation will be used for the wrist, P_5, as visible from frames 0 and 1, respectively, to determine 1. It would seem that the rotation from frame 0-to-1, θ1, would be equivalent to the difference between 0-to-5 and 1-to-5 rotations. Adding formal language and the symbols from Figure 4.4 results in:

$$ v_{0\to1} = v_{0\to5} - v_{1\to5} $$

$$ v_{0\to1} = v_{0\to5} + v_{1\to5} $$

$$ \theta_1 = \emptyset_1 + \left(\emptyset_2 + \frac{\pi}{2}\right) \qquad \text{eq. (14)} $$

Analyzing the triangle with sides ${}_{5x}^{0}P$ and ${}_{5y}^{0}P$ will reveal the angle $\emptyset_1$:

$$\emptyset_1 = \operatorname{atan} 2\left({}_{5y}^{0}P, {}_{5x}^{0}P\right) \qquad \text{eq. (15)}$$

The angle $\emptyset_2$ is established by investigating the right most triangle through $\emptyset_2$ as one of the angles. There are lengths on two of the sides $|{}_{5xy}^{0}P|$ and d_4:

By looking at the rightmost triangle that has the angle _2 as one of its angles, the angle $\emptyset_2$ may be obtained. $|{}_{5xy}^{0}P|$ and d_4 are the lengths of two of the sides:

$$\cos(\emptyset_2) = \frac{d_4}{\left|{}_{5xy}^{0}P\right|}$$

$$\emptyset_2 = \pm \operatorname{acos}\left(\frac{d_4}{\left|{}_{5xy}^{0}P\right|}\right)$$

$$\emptyset_2 = \pm \operatorname{acos}\left(\frac{d_4}{\sqrt{{}_{5x}^{0}P^2 + {}_{5y}^{0}P^2}}\right) \qquad \text{eq. (16)}$$

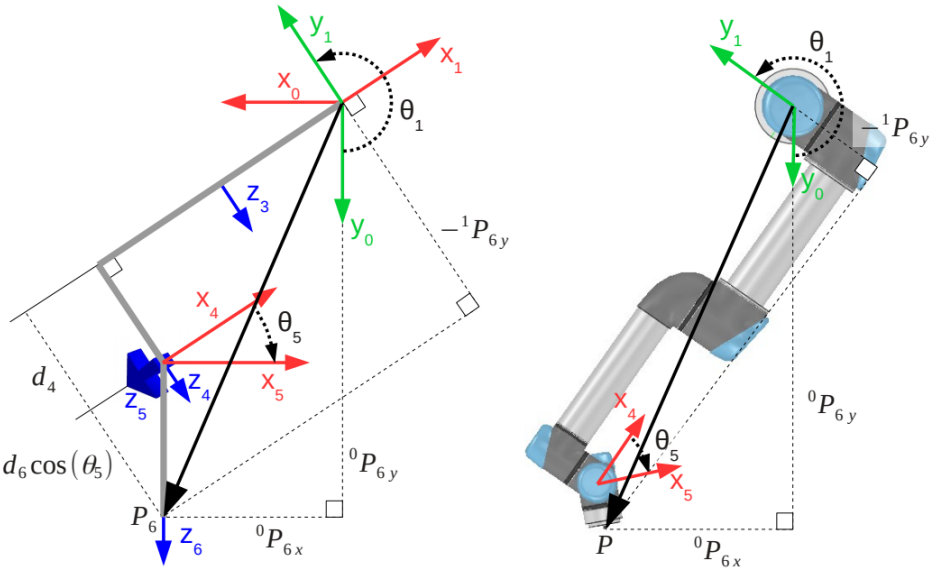
The needed angle θ_1 can now be originate basically as:

$$\theta_1 = \operatorname{atan} 2\left({}_{5y}^{0}P, {}_{5x}^{0}P\right) \pm \operatorname{acos}\left(\frac{d_4}{\sqrt{{}_{5x}^{0}P^2 + {}_{5y}^{0}P^2}}\right) + \frac{\pi}{2} \qquad \text{eq. (17)}$$

The two answers indicate whether the shoulder is "left" or "right." Figure 4.5 depicts the robot from the top view once more, this time incorporating frame 6.

Figure 4.5

Robot from Top View Perspective Including Frame 6



By seeing that ${}^1_{6y}P$, or the y-component of ${}^1_6P$, only depends on θ_5, we can use this information to determine 5. In figure 4.5, if we go backwards from y_1, we can see that ${}^1_6P$ is a rotation of ${}^0_6P$ around z_1:

$$ {}^0_6P = {}^0_1R \,.\, {}^1_6P $$

$$ {}^0_6P = {}^0_1R \,.\, {}^1_6P $$

$$ \begin{bmatrix} {}^1_{6x}P \\ {}^1_{6y}P \\ {}^1_{6z}P \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} {}^0_{6x}P \\ {}^0_{6y}P \\ {}^0_{6z}P \end{bmatrix} $$

$$ \begin{bmatrix} {}^1_{6x}P \\ {}^1_{6y}P \\ {}^1_{6z}P \end{bmatrix} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \begin{bmatrix} {}^0_{6x}P \\ {}^0_{6y}P \\ {}^0_{6z}P \end{bmatrix} $$

$$ {}^1_{6y}P = {}^0_{6x}P \,.\, (-\sin\theta_1) + {}^0_{6y}P \,.\, \cos\theta_1 \qquad \text{eq. (18)} $$

By combining equation (17) and (18), we can eliminate ${}^1_{6y}P$ and express θ_5 only using known values:

Combining equations (17) and (18), we can get rid of ${}^1_{6y}P$ and only use known values to express θ_5:

$$-d_4 - d_6 \cos\theta_5 = {}_{6x}^{0}P(-\sin\theta_1) + {}_{6y}^{0}P\cos\theta_1$$

$$\cos\theta_5 = \frac{{}_{6x}^{0}P\sin\theta_1 - {}_{6y}^{0}P\cos\theta_1 - d_4}{d_6}$$

$$\theta_5 = \pm\operatorname{acos}\left(\frac{{}_{6x}^{0}P\sin\theta_1 - {}_{6y}^{0}P\cos\theta_1 - d_4}{d_6}\right) \qquad \text{eq. (19)}$$

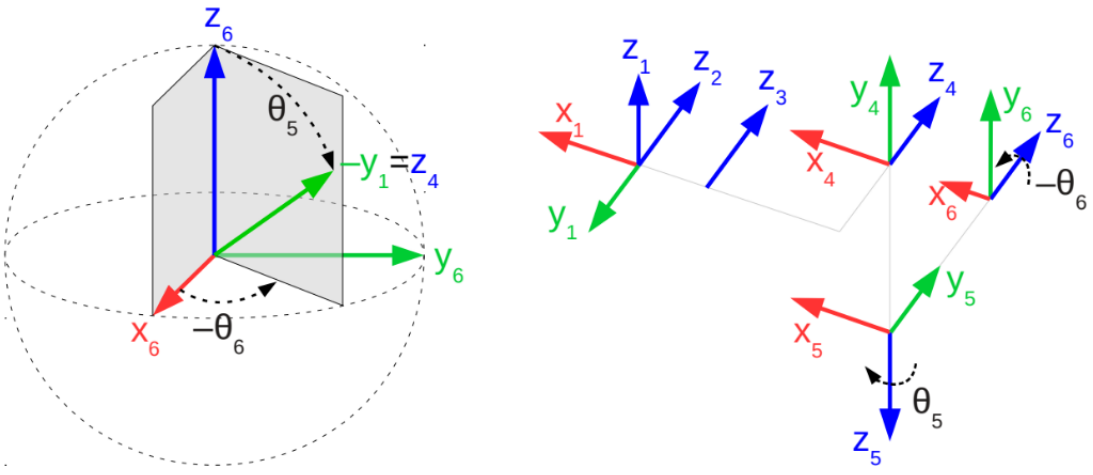
Once more, there are two options. These signify that the wrist is, respectively, "up" or "down." This can be understood logically as follows: The end-effector may be in the same location but with the wrist turned due to the joint sum $(\theta_2 + \theta_3 + \theta_4)$. Then, by using θ_6, the orientation can be "adjusted." Also keep in mind that a solution is considered to exist if and only if the magnitude of the value contained within acos is not larger than 1, or $\left|{}_{6y}^{1}P - d_4\right| \leq |d_6|$.

To determine θ_6 is need to examined y_1 seen from frame 6; ${}_{1}^{6}\hat{Y}$. That axis will (ignoring translations) always be parallel to ${}_{2,3,4}^{6}\hat{Z}$, as illustrates in Figure 4.6 b. Consequently, the dependencies will be only on θ_5 and θ_6. It can be changed that $-{}_{1}^{6}\hat{Y}$ can be designated using sphere-shaped coordinates, where Azimuth is $-\theta_6$ and the polar angle is θ_5; which can be seen Figure 4.6 a.

We look at y 1 as observed from frame 6 to calculate θ_6; ${}_{1}^{6}\hat{Y}$. As seen in Figure 4.6 b, this axis will always be parallel to ${}_{2,3,4}^{6}\hat{Z}$ (ignoring translations). Therefore, only θ_5 and θ_6 will be required. In reality, spherical coordinates can be used to define $-{}_{1}^{6}\hat{Y}$, where Azimuth is $-\theta_6$ and the polar angle is θ_5; see Figure 4.6 a.

Figure 4.6

Azimuth and Polar Angles are Used to Express in Spherical Coordinates



(a) (b)

Notes: (a)Polar Angles for $-{}_1^6\hat{Y}$; (b) Reference View of the Relevant Frames

Spherical coordinates $-{}_1^6\hat{Y}$ are translated to Cartesian coordinates as follows:

$$-{}_1^6\hat{Y} = \begin{bmatrix} \sin\theta_5 \cos(-\theta_6) \\ \sin\theta_5 \cos(-\theta_6) \\ -\cos\theta_5 \end{bmatrix}$$

$${}_1^6\hat{Y} = \begin{bmatrix} -\sin\theta_5 \cos(\theta_6) \\ \sin\theta_5 \cos(\theta_6) \\ -\cos\theta_5 \end{bmatrix} \qquad \text{eq. (20)}$$

In equation (13) we could isolate θ_6 and have an expression θ_6 in relation to ${}_1^6T$. We want an expression of θ_6 all the way from ${}_0^6T$. To get this, we identify that ${}_1^6\hat{Y}$ is given as a rotation of θ_1 in the x, y plane of frame 0 very similar to equation (18)):

We could separate θ_6 from equation (13) and have an expression for θ_6 in relation to ${}_1^6T$. We desire an expression of θ_6 starting at ${}_0^6T$. In order to obtain this which is quite close to equation (18), we recognize that ${}_1^6\hat{Y}$ is given as a rotation of θ_1 in the x, y plane of frame 0:

$${}_1^6\hat{Y} = {}_0^6\hat{X} \,.\, (-\sin\theta_1) + {}_0^6\hat{Y} \,.\, \cos\theta_1$$

$$ {}^{6}_{1}\hat{Y} = \begin{bmatrix} -{}^{6}_{0x}\hat{X} \,.\sin\theta_1 + {}^{6}_{0x}\hat{Y} \,.\cos\theta_1 \\ -{}^{6}_{0y}\hat{X} \,.\sin\theta_1 + {}^{6}_{0y}\hat{Y} \,.\cos\theta_1 \\ -{}^{6}_{0z}\hat{X} \,.\sin\theta_1 + {}^{6}_{0z}\hat{Y} \,.\cos\theta_1 \end{bmatrix} \qquad \text{eq. (21)} $$

Equating the first two entries of equations (18) and (19) give:

$$ \begin{Bmatrix} -\sin\theta_5 \cos\theta_6 = -{}^{6}_{0x}\hat{X} \,.\sin\theta_1 + {}^{6}_{0x}\hat{Y} \,.\cos\theta_1 \\ \sin\theta_5 \sin\theta_6 = -{}^{6}_{0y}\hat{X} \,.\sin\theta_1 + {}^{6}_{0y}\hat{Y} \,.\cos\theta_1 \end{Bmatrix} $$

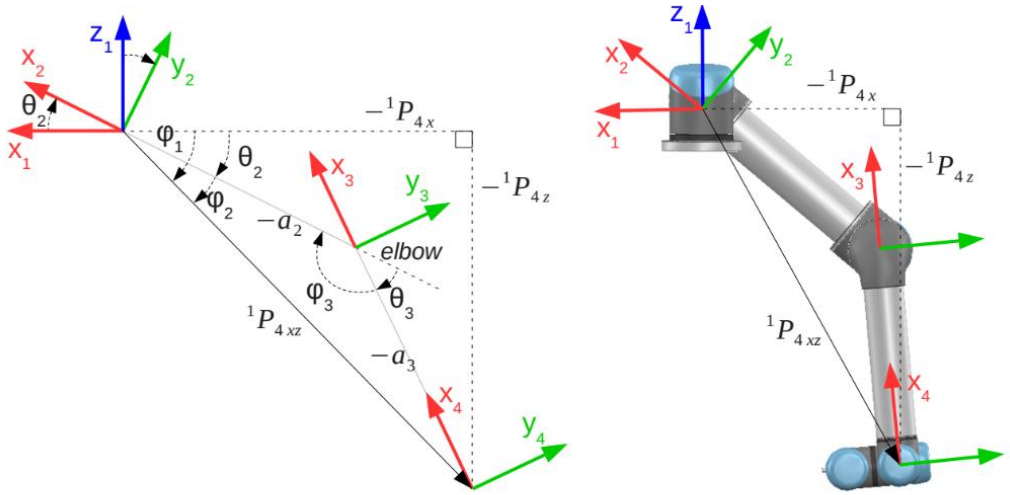
$$ \begin{Bmatrix} \cos\theta_6 = \frac{{}^{6}_{0x}\hat{X} \,.\sin\theta_1 + {}^{6}_{0x}\hat{Y} \,.\cos\theta_1}{\sin\theta_5} \\ \sin\theta_6 = \frac{-{}^{6}_{0y}\hat{X} \,.\sin\theta_1 + {}^{6}_{0y}\hat{Y} \,.\cos\theta_1}{\sin\theta_5} \end{Bmatrix} $$

$$ \theta_6 = \operatorname{atan}2\left(\frac{-{}^{6}_{0y}\hat{X} \,.\sin\theta_1 + {}^{6}_{0y}\hat{Y} \,.\cos\theta_1}{\sin\theta_5}, \frac{{}^{6}_{0x}\hat{X} \,.\sin\theta_1 + {}^{6}_{0x}\hat{Y} \,.\cos\theta_1}{\sin\theta_5}\right) \qquad \text{eq. (22)} $$

If the denominator sin θ_5 = 0, this solution is unknown. The joint axes (2, 3, 4 and 6) are lined up in this instance (as in Figure 4.7 b). Degrees of freedom are "too many" in this case. The 6'th joint is no longer necessary because the end-effector (frame 6) may spin around z_6 on its own without being moved by the axes 2, 3, and 4. In this scenario, it is sufficient to set θ_6 to any value. The answer to Equation (22) is also unknown if either or both of the numerators are 0. The answer to Equation (22) is also unknown if either or both of the numerators are 0. If so, sin θ_5 must also be 0 and the circumstance remains the same. Examining both sides of Equation will demonstrate this equation (21). We look at the final three joints' axes (2, 3, and 4). Their joint axes are all parallel, as you can see. They work as a unit to form a planar 3R-manipulator, as shown in Figure 4.7.

Figure 4.7

3R Planar Manipulator Constitution of Joint 2, 3, and 4



Being aware of ${}^{0}_{1}T$, ${}^{4}_{5}T$, and ${}^{5}_{6}T$ at this point allows us to focus on ${}^{1}_{4}T$ (frame 4 in respect to frame 1) only. In Figure 4.7 a, x, z-plane of frame 1, this transformation is shown. The image makes it obvious that just θ_3 or similarly $\emptyset_3$ determines the length of the translation $|{}^{1}_{4xz}P|$. The law of cosine can be used to get the angle $\emptyset_3$:

$$\cos\emptyset_3 = \frac{(-a_2)^2+(-a_3)^2-\left|{}^{1}_{4xz}P\right|^2}{2(-a_2)(-a_3)} = \frac{{a_2}^2+{a_3}^2-\left|{}^{1}_{4xz}P\right|^2}{2a_2a_3} \qquad \text{eq. (23)}$$

The relationship between $\cos\emptyset_3$ and $\cos\theta_3$ is:

$$\cos\theta_3 = \cos(\pi-\emptyset_3) = -\cos(\emptyset_3) \qquad \text{eq. (24)}$$

Combining equation (23) and (24) give:

$$\cos\theta_3 = -\frac{{a_2}^2+{a_3}^2-\left|{}^{1}_{4xz}P\right|^2}{2a_2a_3}$$

$$\theta_3 = \pm\operatorname{acos}\left(\frac{\left|{}^{1}_{4xz}P\right|^2-{a_2}^2-{a_3}^2}{2a_2a_3}\right) \qquad \text{eq. (25)}$$

If the argument of acos falls within [1; 1], then solutions exist for the equation 3. This is equivalent to $|{}^{1}_{4xz}P| \in [|a_2 - a_3|; |a_2 + a_3|]$ and can be demonstrated. There are often two different solutions for every problem if one exists. These signify "elbow up"

and "elbow down," respectively. The angle θ_2 can be calculated by taking $\emptyset_1 - \emptyset_2$. Figure 4.7a and the atan2 and sine relations can be used to find each of these:

$$\emptyset_1 = \text{atan}\,2(-{}_{4z}^{1}P, -{}_{4x}^{1}P)$$

$$\frac{\sin\emptyset_2}{-a_3} = \frac{\sin\emptyset_3}{\left|{}_{4xz}^{1}P\right|}$$

$$\emptyset_2 = \text{asin}\left(\frac{-a_3\sin\emptyset_3}{\left|{}_{4xz}^{1}P\right|}\right) \qquad \text{eq. (26)}$$

By seeing that sin 3 = sin (180 3) = sin 3, we can replace 3 with 3. Adding the equations together results in:

$$\theta_2 = \emptyset_1 - \emptyset_2 = \text{atan}\,2(-{}_{4z}^{1}P, -{}_{4x}^{1}P) - \text{asin}\left(\frac{-a_3\sin\emptyset_3}{\left|{}_{4xz}^{1}P\right|}\right) \qquad \text{eq. (27)}$$

The angle between ${}_{4y}^{3}\hat{X}$ to X_4 measured about Z_4 is referred to as the residual angle θ4 (cf. Equation (3) on page 3). Thus, using the first column of the final transformation matrix, ${}_{4y}^{3}X$, the following may be simply deduced:

$$\theta_4 = \text{atan}\,2\left({}_{4y}^{3}\hat{X}, {}_{4x}^{3}\hat{X}\right) \qquad \text{eq. (28)}$$

4.2.1.3 **Robotic Path Simulation and Off-Line Programming.** The created virtual world allows to conduct dynamics 3D simulations for robots and components. The layout of the robots according to part localization can be calculated by performing robotics arm reach checks to those desired locations. By using 3D navigation and graphical programming, developing 3D robotics paths and processes for robotics arms in the virtual world to generate the robotics program (.urp) file to download the final program to the physical robots. Realistic robot simulation (RRS) allows to calculate accurate cycle time between two robotics arms to get better collaboration and time management between two robots.

In Figure 4.8, The red box is the 3D viewer where you can see your production layout, the green box represents the operation tree where you can store your developed program, the yellow box is the object tree where you can import and export the objects that you want to use in the virtual world and the last but not least the blue box represents the robotics path editor and sequence editor for the whole process. The green box in Figure 4.9 represents the collision viewer and editor window in Tecnomatix process simulate digital twin software.

Figure 4.8

The Software Layout of the Digital Twin Program

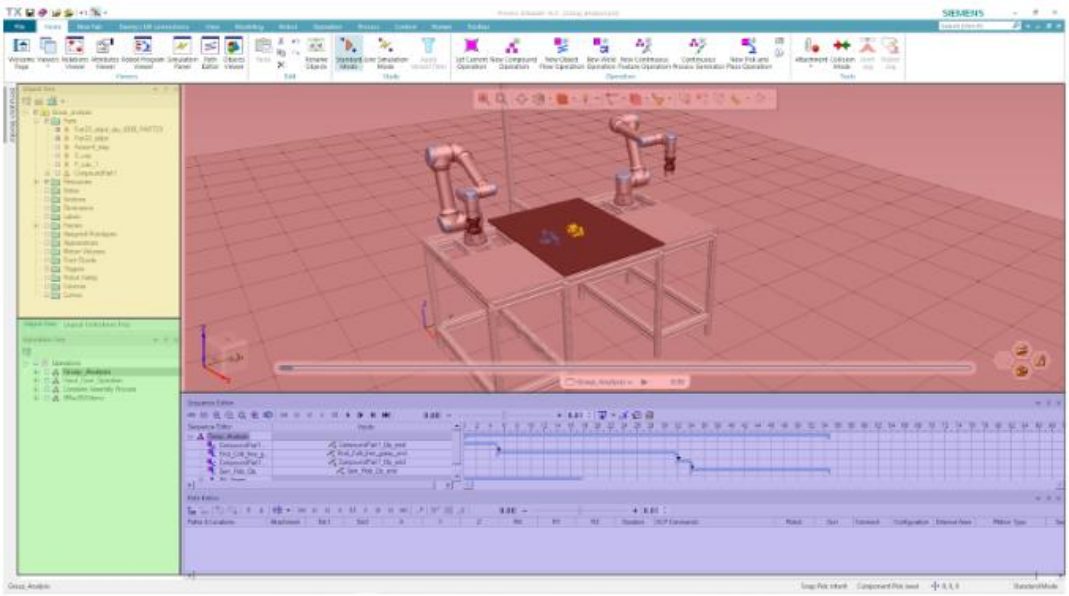
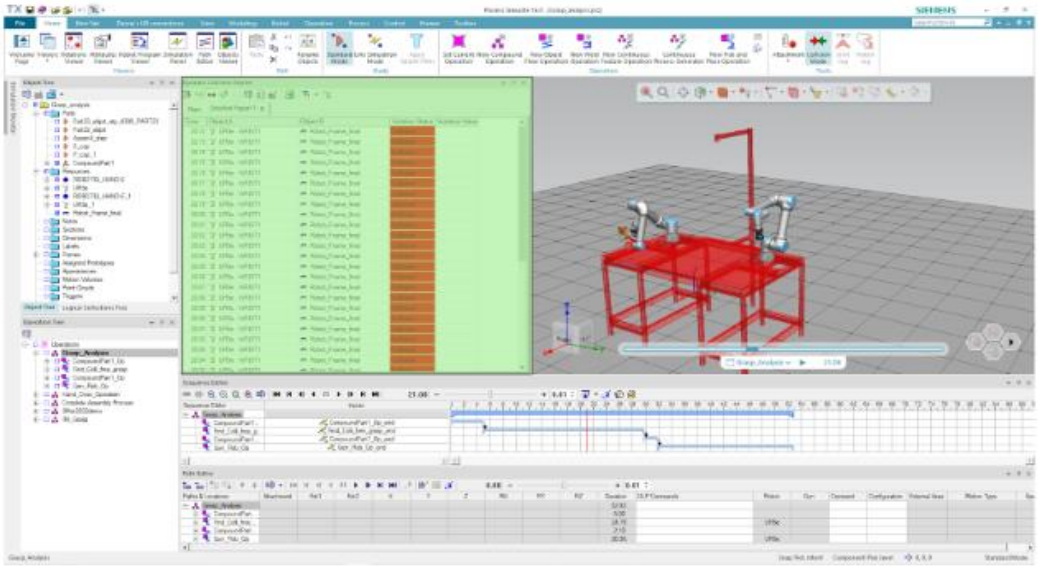


Figure 4.9

Software Implementation Collision Viewer and Editor Window

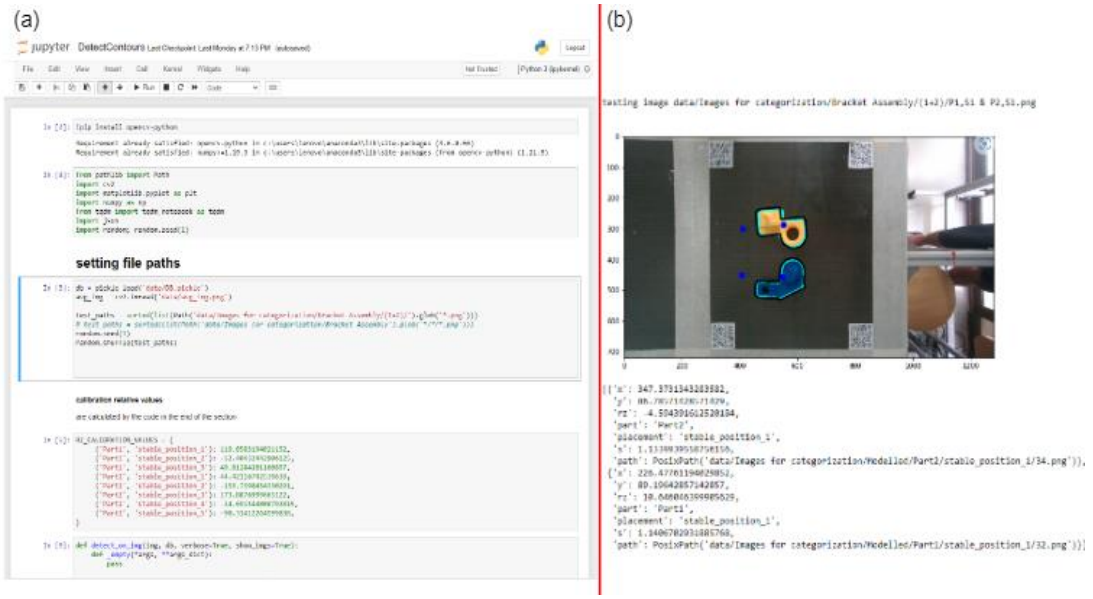


4.3 Part's Detection Program Development

In order to validate the developed concept, an image processing program was developed as “Part recognition, Part’s coordination collection and communication” by using Jupyter Notebook. The program consists of 3 parts. As shown in Figure 4.10 (a), the main program for part recognition and collection of part’s coordinates are partially developed in the Jupyter Notebook. The program layout after part recognition and coordinate collection is shown in Figure 4.10 (b).

Figure 4.10

Main Program Layout for Part Recognition and Coordinate Collection

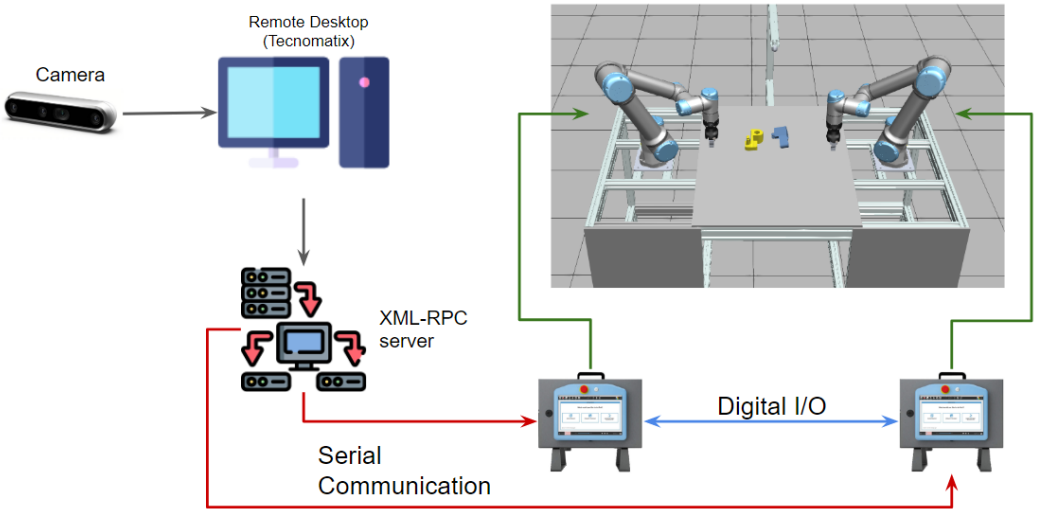


Notes: (a) Program Layout Before Executing; (b) Program Layout After Executing

After that, the communication pipeline program is developed for each robot by establishing an XML_RPC server in order to transfer the data to the robot and vice versa. The complete communication setup between both robots, camera and pc is shown in Figure 4.11.

Figure 4.11

Complete XML_RPC Communication Setup



CHAPTER 5
IMPLEMENTATION AND RESULTS

5.1 System Operation

The Co-MRAS concept was established in session 3.2, and the implementation procedure followed the created algorithm as indicated in session 4.1 with the aid of Tecnomatix Process Simulate (TPS) digital twin software, developed software (Parts' detection and coordinates collection), XML_RPC server and virtual machine software. The XML RPC server must first be set up before the physical robotics arms can be connected to the appropriate communication programs. The appropriate CAD model is then split up into several files in the (.cojt) format so that TPS can build the assembly process.

The collision grasps for initial and final state of the assembly operation files obtained from the collision checking module in TPS software contain many kinds of collisions and the operator needs to identify the collision free grasps and develop the predefined collision free robotics paths for different configuration in TPS Software. The developed software (Part's detection and coordinates collection) is used to identify the different parts in different initial configurations and calculate the parts' coordinate according to the reference world coordinates. The collected coordinates are transferred to the real robots by using XML_RPC server.

5.2 Implementation

The implementation procedure was carried out using six components, each of which had a distinct size and shape as indicated in Table 5.1 with illustrations, in order to validate the idea of Co-MRAS, dimensions of peg and hole 3D components and numbers of stable orientation placement on the work table. The proposed method is being validated with two different assembly cases, which have complex parts' shape and different initial stable placement on the worktable.

First assembly case consists of two mechanical components with a double peg in hole assembly to be handled by two robotics arms and the other assembly case consists of four different components with a peg in hole assembly for three times to ensure the precision mating during the assembly process.

Table 5.1

Assembly Components Samples for Implementation

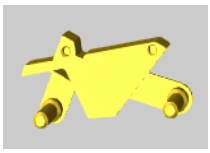
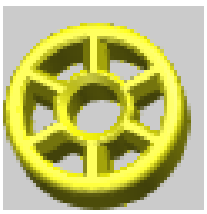
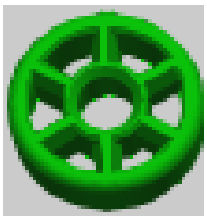
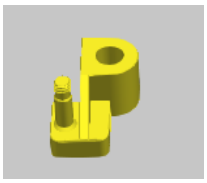
No	Component name	Figures	Dimension	Number of stable Placement
1	Cycle Body Frame		110mm x 60mm x 40mm	3
2	Cycle Body Connector		110mm x 60mm x 20mm	2
3	Wheel		43mm x 43mm x 17mm	1
4	Wheel		43mm x 43mm x 17mm	1
5	Mechanical Bracket Part 1		120mm x 90mm x 75mm	3
6	Mechanical Bracket Part 2		120mm x 90mm x 80mm	5

Figure 5.1 represents the assembly stages of two components in one random configuration. Stage (a) shows the beginning of the assembly process, where the left robot picks up the yellow part. Stage (b) shows the right robot picking up the blue part and Stage (c) shows the step of peg in hole assembly by two robotics arms collaboratively.

Figure 5.1

No Hand Over Assembly Process Random Configuration

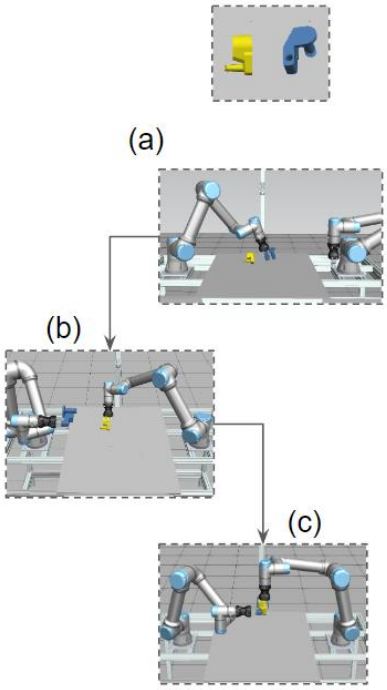


Figure 5.2 represents the assembly stages of two components in one random configuration which is that the assembly components are needed to hand over to manipulate the orientation of the component. Stage (a) shows the beginning of the assembly process, where the left robot picks up the yellow component. Stage (b) shows the left robot hand over the component to the right robot. Stage (c) illustrates the left robot picking up the blue part and Stage (d) and (e) describes the two robots performing the peg in hole assembly process.

Figure 5.2

Hand Over Assembly Process Random Configuration

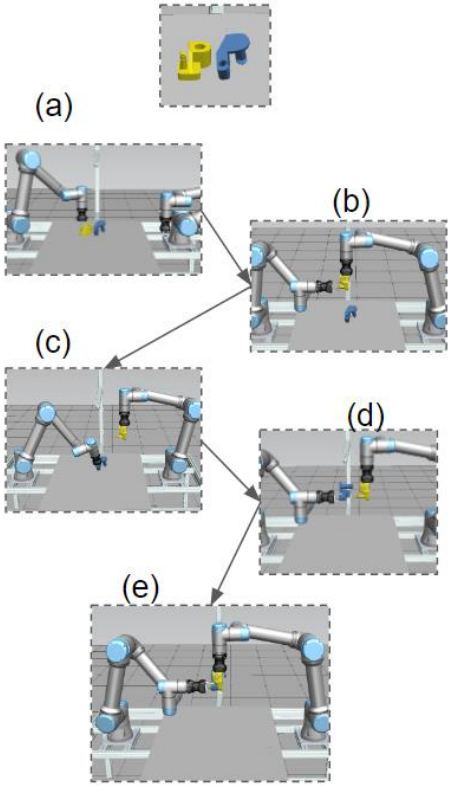
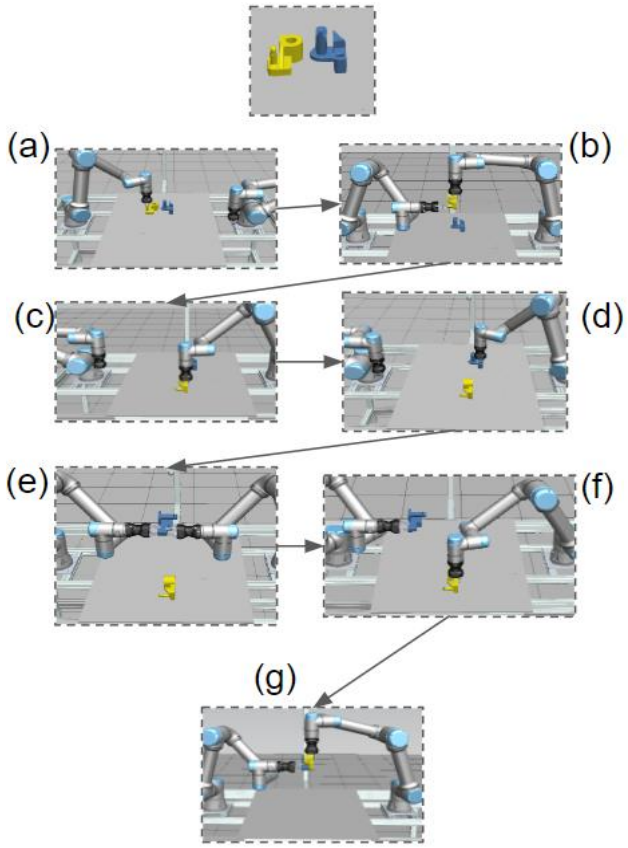


Figure 5.3 represents the assembly process of two components which are in one random configuration in which the robot needed to collaborate with another robot to reorient the assembly components. As aforementioned, in Figure 5.1 and Figure 5.2 the first 3 stages are same with Figure 5.2 process and stage (c) represents the right robot need to place the component onto the work table in order to pick up the blue part and need to collaborate with the left arm to reorient the component to the desire orientation which are presented in Stage (d) and (e). Stage (f) shows the right arm picking up the yellow part from the work table back to perform peg in hole assembly operations as illustrated in Stage (g).

Figure 5.3

Handover Process and Ideal Position Process



5.3 Result

The outcomes of the testing and validation of the six sample components' implementation processes are presented in this chapter. According to the testing samples, for the image coordinated feedback accuracy is depending on the placement of the samples on the work table because of the camera resolution and distorted areas captured by the camera. Even the undistorted camera calibration is conducted to solve the distortion error, the shifting results in x and y coordinates of each assembly components are found in different location on the worktable.

Therefore, the red rectangle box which size is (320x214) mm is created in order to get the accurate parts' coordinates from the image processing algorithm which means if the tested sample coordinates are outside of the red rectangle box, the accuracy results of each sample is reduced by 5 percentage error value because of the camera resolution and distorted value and moreover that can be affect the final outcome of the whole

assembly process. (See in Figure 5.4). Figure 5.5 illustrates the recognition precision comparison between inside and outside of the box.

The results of the parts' coordinates and stable placement numbers are collected and send the coordinates to the robots to reach the desired grasps locations using the developed XML_RPC protocol. The results of the parts' coordinates and physical outcome the two components mechanical bracket assembly case by robotics arms are being presented in the Table 5.2. Table 5.3 shows the final outcome of the robotics arm and parts' coordinates results of the four components cycle assembly.

Figure 5.6 illustrates the digital twin assembly process; the virtual and physical world are presented as bracket assembly process case. The pre-assembly process planning in virtual world is completely the same with physical actual assembly process. The digital twin program for both virtual and physical world is the same for both robots.

Table 5.2

Result of Mechanical Bracket Assembly Case and Parts' Coordinates

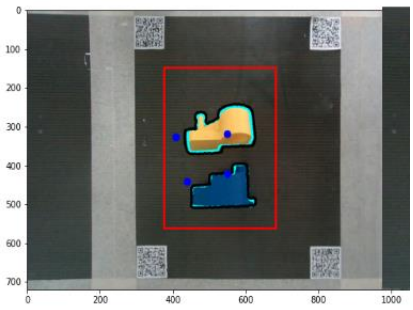
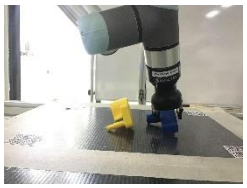
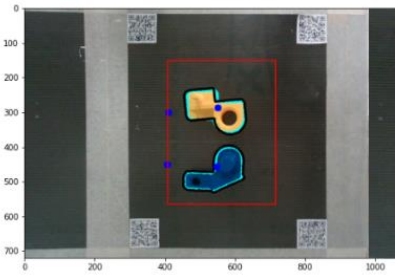
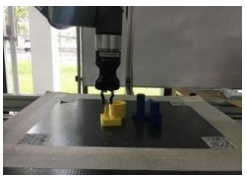
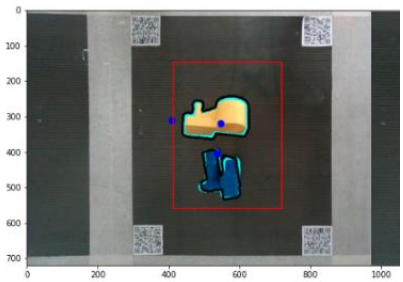
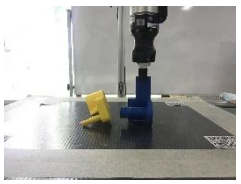
Image Analysis	Parts' Coordinates	Result
	'x': 247.433, 'y': 87.589, 'rz': 1.411, 'part': 'Part1', 'placement': 'stable_position_2'	
	'x': 340.119, 'y': 112.500, 'rz': -0.530, 'part': 'Part2', 'placement': 'stable_position_4'	
	'x': 226.448, 'y': 87.196, 'rz': 10.646, 'part': 'Part1', 'placement': 'stable_position_1'	
	'x': 347.373, 'y': 86.786, 'rz': -4.594, 'part': 'Part2', 'placement': 'stable_position_1'	
	'x': 234.537, 'y': 88.293, 'rz': -5.730, 'part': 'Part1', 'placement': 'stable_position_2'	
	'x': 310.299, 'y': 191.250, 'rz': -6.981, 'part': 'Part2', 'placement': 'stable_position_5'	

Table 5.3

Result of Cycle Assembly Case and Parts' Coordinates

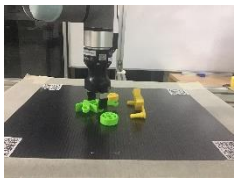
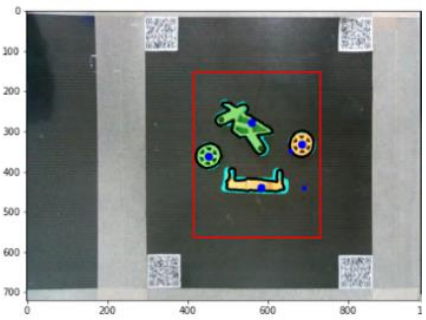
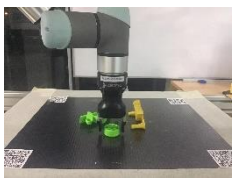
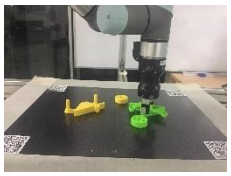
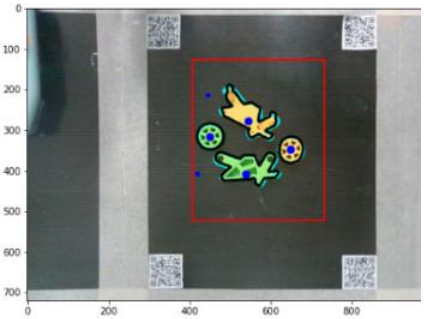
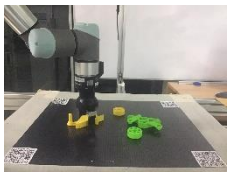
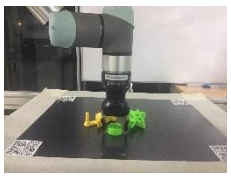
Image Analysis	Parts' Coordinates	Result
	'x': 208.746, 'y': 209.732, 'rz': 120.117, 'part': 'Cycle Body (Green)', 'placement': 'stable_position_1'	
	'x': 338.507, 'y': 229.017, 'rz': 92.732, 'part': 'Cycle Body (Yellow)', 'placement': 'stable_position_4'	
	'x': 252.268, 'y': 310.178, 'rz': 46.453, 'part': 'Wheel (Yellow)', 'placement': 'stable_position_1'	
	'x': 276.447, 'y': 123.750, 'rz': -88.532, 'part': 'Wheel (Yellow)', 'placement': 'stable_position_1'	
	'x':208.746, 'y': 209.732, 'rz': 120.117, 'part': 'Cycle Body (Green)', 'placement': 'stable_position_1'	
	'x': 207.940, 'y': 197.678, 'rz': 29.862, 'part': 'Cycle Body (Yellow)', 'placement': 'stable_position_1'	
	'x': 264.358, 'y': 281.250, 'rz': -46.542, 'part': 'Wheel (Yellow)', 'placement': 'stable_position_1'	
	'x': 239.373, 'y': 121.339, 'rz': -86.455, 'part': 'Wheel (Green)', 'placement': 'stable_position_1'	

Figure 5.4

Percent Error Comparison Inside and Outside of Rectangle Box

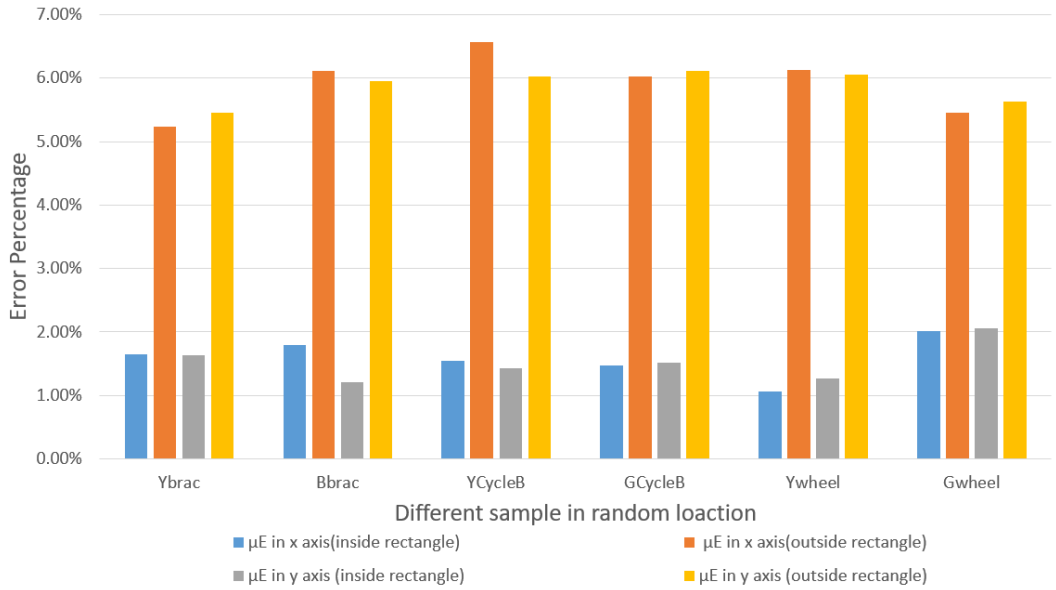


Figure 5.5

Part Recognition Precision Comparison Inside and Outside of Rectangle Box

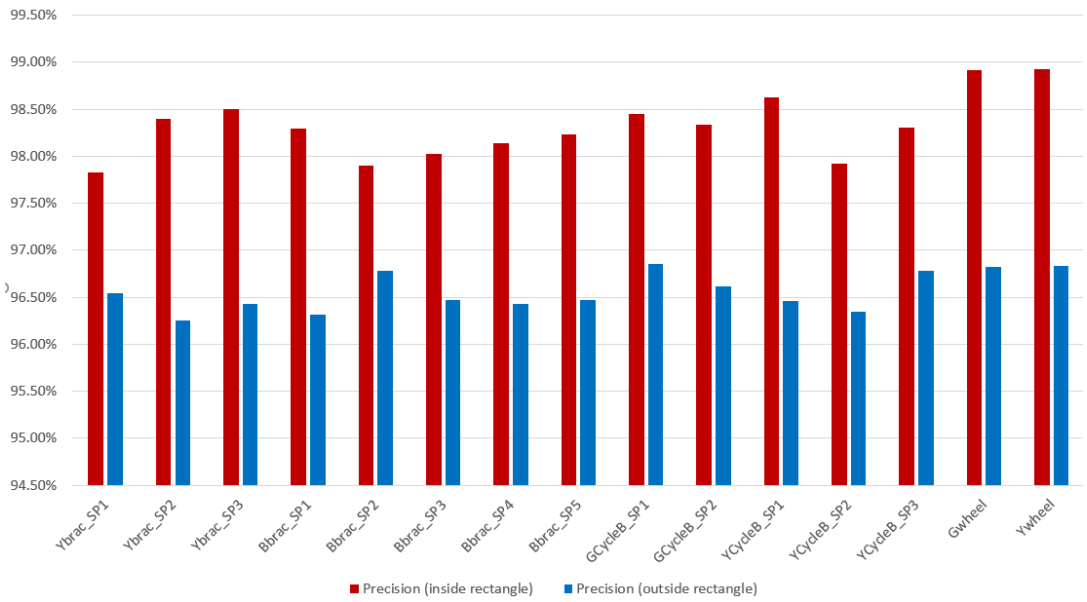
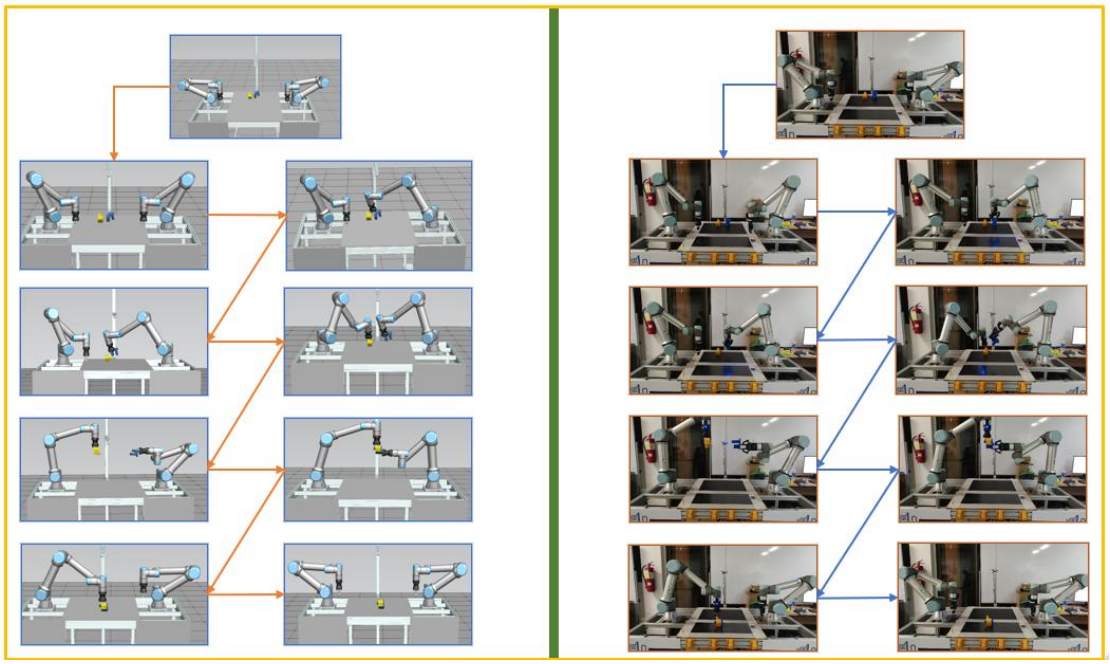


Figure 5.6

Virtual World and Physical World Bracket Assembly Case



CHAPTER 6
CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

The major goal of this research is to create an autonomous collaborative multi-robotics assembly system (Co-MRAS) that will enable better handling of complicated 3D components in various initial configuration setups that are chosen at random. This is accomplished via the successfully developed Co-MRAS concept, which includes initial and final grasp's location verification, Collision free robot's paths generation, Parts' coordinates collection, Final program generation and Communication protocol between robots and computers.

An algorithm is being developed within the TPS program to execute the idea of Co-MRAS and allows for verification of initial and final grasps locations, identification of collision and generation of collision free motion for multiple robotics arms. Moreover, this paper develops to an image processing program to collect the parts' information and coordinates comparing with the world coordinates from 3D space.

6.2 Recommendations

A new digital twin approach for autonomous collaborative multi robotics assembly system (Co-MRAS) is proposed in this paper to generate collision free complete robotics programs for multiple robotics arms to handle the complex shape parts and non-repetitive tasks for customized mechanical parts in 3D space. The proposed digital twin approach can handle the different initial random stable placement on the work table during assembly process by using the verification of initial and final grasp's location in virtual world simulation environment.

The resulting sequences can be used by a motion planner to plan the motion for both robotics arms. Currently, the approach is established by being separated into different programs to get the complete result. In the future, the proposed step by step approach can be improved by developing the integrated software in order to use all the approaches in one program platform.

REFERENCES

Baykal, I. C. (2018, September). Inspection of Screw Holes on Machine Parts Using Robot Vision. In 2018 International Conference on Artificial Intelligence and Data Processing (IDAP) (pp. 1-5). IEEE.

Chang, W. C., Weng, Y. H., Tsai, Y. H., & Chang, C. L. (2011, June). Automatic robot assembly with eye-in-hand stereo vision. In 2011 9th World Congress on Intelligent Control and Automation (pp. 914-919). IEEE.

Chang, W. C., & Wu, C. H. (2017, April). Automated USB peg-in-hole assembly employing visual servoing. In 2017 3rd International Conference on Control, Automation and Robotics (ICCAR) (pp. 352-355). IEEE.

Chen, H., Wan, W., & Harada, K. (2021, August). Planning to build soma blocks using a dual-arm robot. In 2021 IEEE International Conference on Development and Learning (ICDL) (pp. 1-7). IEEE.

Chryssolouris, G., Mavrikios, D., Papakostas, N., Mourtzis, D., Michalos, G., & Georgoulias, K. (2009). Digital manufacturing: history, perspectives, and outlook. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 223(5), 451-462.

Da Xu, L. (2020). Industrial information integration–An emerging subject in industrialization and informatization process. Journal of Industrial Information Integration, 17, 100128.

Dobashi, H., Hiraoka, J., Fukao, T., Yokokohji, Y., Noda, A., Nagano, H., ... & Tanaka, K. I. (2014). Robust grasping strategy for assembling parts in various shapes. Advanced Robotics, 28(15), 1005-1019.

Dogar, M., Spielberg, A., Baker, S., & Rus, D. (2019). Multi-robot grasp planning for sequential assembly operations. Autonomous Robots, 43(3), 649-664.

Huang, Y., Zhang, X., Chen, X., & Ota, J. (2017). Vision-guided peg-in-hole assembly by Baxter robot. Advances in Mechanical Engineering, 9(12), 1687814017748078.

Jiang, J., Huang, Z., Bi, Z., Ma, X., & Yu, G. (2020). State-of-the-Art control strategies for robotic PiH assembly. Robotics and Computer-Integrated Manufacturing, 65, 101894.

Jiansong, L., Haitao, W., Hu, S., & Ma, C. (2018, June). Study on guide hole detection method for precise assembly based on binocular stereo vision. In CSAA/IET

International Conference on Aircraft Utility Systems (AUS 2018) (pp. 685-688). IET.

Jiménez, P. (2013). Survey on assembly sequencing: a combinatorial and geometrical perspective. Journal of Intelligent Manufacturing, 24(2), 235-250.

Kousi, N., Koukas, S., Michalos, G., & Makris, S. (2019). Scheduling of smart intra–factory material supply operations using mobile robots. International Journal of Production Research, 57(3), 801-814.

Kousi, N., Michalos, G., Aivaliotis, S., & Makris, S. (2018). An outlook on future assembly systems introducing robotic mobile dual arm workers. Procedia CIRP, 72, 33-38.

Krüger, J., Wang, L., Verl, A., Bauernhansl, T., Carpanzano, E., Makris, S., ... & Pellegrinelli, S. (2017). Innovative control of assembly systems and lines. CIRP annals, 66(2), 707-730.

Li, K., Liu, Q., Xu, W., Liu, J., Zhou, Z., & Feng, H. (2019). Sequence planning considering human fatigue for human-robot collaboration in disassembly. Procedia CIRP, 83, 95-104.

Liu, C., Xu, J., Zhao, J., Chen, H., Xi, N., & Chen, K. (2016, July). Non-vector space visual servoing for multiple pin-in-hole assembly by robot. In 2016 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO) (pp. 134-140). IEEE.

Liu, S., Xu, D., Zhang, D., & Zhang, Z. (2014). High precision automatic assembly based on microscopic vision and force information. IEEE Transactions on Automation Science and Engineering, 13(1), 382-393.

Makris, S., Michalos, G., & Chryssolouris, G. (2012). Virtual commissioning of an assembly cell with cooperating robots. Advances in Decision Sciences, 2012.

Mei, B., Zhu, W., Yan, G., & Ke, Y. (2015). A new elliptic contour extraction method for reference hole detection in robotic drilling. Pattern Analysis and Applications, 18(3), 695-712.

Michalos, G., Kousi, N., Makris, S., & Chryssolouris, G. (2016). Performance assessment of production systems with mobile robots. Procedia CIRP, 41, 195-200.

Michalos, G., Makris, S., Papakostas, N., Mourtzis, D., & Chryssolouris, G. (2010). Automotive assembly technologies review: challenges and outlook for a flexible

and adaptive approach. CIRP Journal of Manufacturing Science and Technology, 2(2), 81-91.

Nagarajan, P., Saravana Perumaal, S., & Yogameena, B. (2016, December). Vision based pose estimation of multiple peg-in-hole for robotic assembly. In International conference on computer vision, graphics, and image processing (pp. 50-62). Springer, Cham.

Rodrıguez, I., Nottensteiner, K., Leidner, D., Kaßecker, M., Stulp, F., & Albu-Schäffer, A. (2019). Iteratively refined feasibility checks in robotic assembly sequence planning. IEEE Robotics and Automation Letters, 4(2), 1416-1423.

Sun, C., Liu, Q., & Wu, C. (2016, December). Technique of log piles location based on visual saliency in port. In 2016 International Conference on Industrial Informatics-Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII) (pp. 105-108). IEEE.

Tao, S., & Hu, M. (2017). A contact relation analysis approach to assembly sequence planning for assembly models. Computer-Aided Design and Applications, 14(6), 720-733.

Thomas, U., & Wahl, F. M. (2001, October). A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. In Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180) (Vol. 3, pp. 1458-1464). IEEE.

Tian, G., Zhou, M., & Li, P. (2017). Disassembly sequence planning considering fuzzy component quality and varying operational cost. IEEE Transactions on Automation Science and Engineering, 15(2), 748-760.

Wang, J., & Cho, H. (2008). Micropeg and hole alignment using image moments based visual servoing method. IEEE Transactions on Industrial Electronics, 55(3), 1286-1294.

Xu, F., Wang, S., & Li, B. (2014, June). Industrial robot base assembly based on improved Hough transform of circle detection algorithm. In Proceeding of the 11th World Congress on Intelligent Control and Automation (pp. 2446-2450). IEEE.

Yang, Z., Liu, W., Li, H., & Li, Z. (2018, October). A coaxial vision assembly algorithm for un-centripetal holes on large-scale stereo workpiece using multiple-dof

robot. In 2018 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1-6). IEEE.

Yu, M., Shao, L., Chen, Z., Wu, T., Fan, Q., Mo, K., & Dong, H. (2021). RoboAssembly: Learning Generalizable Furniture Assembly Policy in a Novel Multi-robot Contact-rich Simulation Environment. arXiv preprint arXiv:2112.10143.

Zhang, Z., Zhang, J., & Xu, D. (2013, January). Design of microassembly system and research on coarse-to-fine alignment strategy in combination with active zooming. In 2013 IEEE Workshop on Robot Vision (WORV) (pp. 76-81). IEEE.