Force based robotized assembly – mobile manipulator use case

Duarte Alão Magalhães Silva Moutinho

Master Dissertation

Supervisor: Prof. Germano Veiga (PhD)

U. PORTO
FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Integrated Master in Mechanical Engineering

June 2020

Resumo

Muitas indústrias modernas ainda dependem dos processos tradicionais de assemblagem manual, com as suas típicas limitações de eficiência, qualidade inconsistente do produto final e, em alguns casos, problemas ergonómicos para os trabalhadores. A implementação de estações de assemblagem robotizada com ambiente de trabalho estruturado permite ultrapassar estas limitações, mas só podem ser utilizadas em indústrias de produção em massa devido aos seus elevados custos. Em contrapartida, a implementação de estações flexíveis de assemblagem robotizada necessita da utilização de soluções tecnológicas como controlo de força ou visão artificial para ultrapassar as incertezas posicionais presentes num ambiente de trabalho não estruturado. Com o objetivo de materializar uma estação móvel de assemblagem flexível, a presente dissertação procura desenvolver estratégias baseadas em controlo de força para ultrapassar as incertezas posicionais provenientes do sistema de posicionamento da plataforma móvel.

As capacidades de controlo de força do manipulador utilizado (KUKA LBR iiwa 14R 820) foram exploradas, culminando em duas estratégias de assemblagem distintas. Métodos baseados em estratégias de oscilação são capazes de realizar montagens simples, através da execução de rotinas de "procura cega" de modo a inserir uma determinada peça na correspondente ranhura. Este método não elimina as incertezas presentes no ambiente de trabalho e, como tal, tem de ser executado individualmente para cada assemblagem realizada. Por outro lado, o método baseado em exploração do ambiente complementado por correção de orientação utiliza movimentos para sondar o ambiente de trabalho e eliminar completamente as incertezas presentes. Por esta razão, este método é adequado para estações com várias montagens, uma vez que a rotina só necessita de ser executada uma vez.

Como a implementação do método de exploração do ambiente é dependente da geometria da estação de assemblagem, este método não pode ser diretamente utilizado noutras estações de montagem. Assim, com o objetivo de ultrapassar esta limitação e generalizar a rotina desenvolvida para qualquer operação de assemblagem, foi proposto um acessório externo, composto por duas superfícies ortogonais. Este acessório permite a implementação otimizada deste método em qualquer assemblagem com incertezas posicionais planares. Alternativamente, a geometria de cada estação de montagem pode ser analisada de modo a tentar identificar duas superfícies ortogonais desimpedidas, nas quais os movimentos de sonda possam ser realizados.

A implementação das rotinas desenvolvidas na configuração final do manipulador móvel permitiu analisar o impacto relativo de cada um dos erros de posicionamento da plataforma. Foi concluído que os erros de orientação da plataforma móvel são diretamente responsáveis por grandes erros de translação na montagem e, como tal, devem ser limitados ao máximo. Além disso, como esta correspondência de erros é amplificada pela distância entre o manipulador e o local da montagem, esta distância também deve ser tanto menor quanto possível.

Abstract

Modern industries still heavily rely on manual assembly processes, with its typical shortcomings of low efficiency, inconsistent end product quality and, in some cases, ergonomic problems to the workers. The implementation of dedicated large-scale robotic assembly stations with highly structured work environments is capable of solving these problems but can only be employed in mass production industries due to their large costs. Alternatively, the implementation of flexible robotic assembly stations requires the use of technologies such as force control or computer vision to overcome the positional uncertainties of the present unstructured work environment. In order to achieve a flexible mobile assembly station, the present dissertation aims to develop force-controlled strategies to overcome positional uncertainties arising from the positioning system of the mobile platform.

The inherent compliant capabilities of the used manipulator (KUKA LBR iiwa 14R 820) were explored, culminating on two different assembly strategies. Oscillation based methods are capable of accomplishing simple assemblies by employing blind search routines to fit a certain workpiece into the required slot. These routines do not fully eliminate the present uncertainties; therefore, they must be performed for each assembly task. On the other hand, the method of environment exploration with orientation correction uses probing motions to completely eradicate the uncertainties present on the workstation; thus, it is suitable for executing multiple assemblies on a single station, as the routine only needs to be performed once.

Since the implementation of the latter method is highly reliant on the geometry of the workstation, it cannot be directly employed in other assembly stations. Thus, with the goal of generalizing the developed methods, an external accessory with two orthogonal surfaces was proposed. This accessory allows the optimized implementation of the environment exploration with orientation correction method on any assembly station with planar positional uncertainties. Alternatively, the geometry of each station can be examined in order to identify two reachable, orthogonal surfaces that can be used for this routine.

The implementation of the developed routines on the mobile manipulator setup revealed the importance of each present error. In fact, since orientation errors on the mobile platform are responsible for large translation errors on the assembly workspace, they must be limited at all costs. Furthermore, the distance between the manipulator and the station should also be minimized since it amplifies this error conversion.

Acknowledgements

The process of writing this dissertation has been both gratifying and challenging. Without the assistance and guidance of the following people, this would not have been achievable. For that, I own them my sincere gratitude.

First and foremost, I am grateful to my supervisor, Professor Germano Veiga, not only for his orientation and guidance, which were essential to achieve the proposed goals, but also for all his patience and support throughout the semester.

I would also like to acknowledge every colleague that I had the pleasure of working with at INESC TEC's CRIIS department. Their knowledge, assistance and cooperation were vital for the developed study and their welcoming nature and support were crucial for my effortless integration.

Finally, I would like to thank all my friends and family for their encouragement and unparalleled support. Without you, this would not have been possible. Thank you all.

Table of contents

List of figures

List of tables

1. Introduction

1.1. Background and motivations

Robotics is an interdisciplinary field of computer science and engineering that deals with the design, construction and operation of not only robots, but also computer systems for their control, information processing and sensory feedback. It aims to design intelligent and autonomous machines that can safely assist humans.

Robots are highly sought after by many industries to reduce costs or boost productivity. They can be used to replace workers by replicating human actions or to perform tasks unsuited for workers, such as operating in dangerous environments (e.g. radioactive, high heat, underwater).

Industrial robots are traditionally used for welding, painting, material handling and other non-contact or limited contact processes. These tasks are simple and usually do not require robot-environment interaction (unconstrained motions). Technological innovations such as machine vision and force control are expanding the use of robotics to other manufacturing processes, as they allow the implementation of constrained motions, i.e. motions with interaction between the manipulator and its environment. Grinding, cutting and assembly are examples of manufacturing processes performed with constrained motions [1]–[3].

Some industries have been evolving towards mass customization, instead of the traditional mass production. This trend leads to more product variations, shorter life cycles and smaller batch sizes. Thus, it may no longer be economically viable to invest in dedicated assembly equipment for a given product configuration and it may be advantageous to invest in flexible solutions based in robotic systems. These solutions can assemble multiple product configurations due to its ease of programming and task reconfiguration. Furthermore, these systems are more capable of overcoming parametric uncertainties and can safely work alongside humans [4]–[6].

Even though the demand for flexible robotic assembly stations is increasing, its present industrial applications are limited as it is still a developing technology with major shortcomings for large-scale implementation. As such, the main scope of this topic is still heavily based on research and development, with numerous research facilities around the world collaborating to develop this technology [2].

1.2. Project exposition and goals

The Institute for Systems and Computer Engineering, Technology and Science, or INESC TEC, is a private non-profit Portuguese research institution. Its main mission is to advance science and technology, enabling science-based innovation through the transfer of new knowledge and technologies to industry, services and public administration.

Since 2017, INESC TEC is coordinating a European H2020 project with focus on factories of the future, which intends to develop scalable production systems. This project is named ScalABLE 4.0 and seeks the creation of highly flexible factories, through the integration of robots and factory floor equipment.

The present study was conducted at INESC TEC, with the collaboration of the ScalABLE 4.0 project, in which a flexible robotic assembly system was developed. This system consists of a mobile robot transporting a collaborative manipulator. The former is responsible for the transport and correct positioning of the manipulator, whereas the latter is liable for the successful completion of the assembly tasks.

The focus of this study is to explore the capability of force sensing as a way of overcoming positional uncertainties in robotic assembly operations, i.e. using force control to surpass the positional uncertainties of the mobile robot and successfully complete the assembly task. The main goals of the developed work are as follows:

i. Development of routines capable of overcoming positional uncertainties using the inherent compliant control capabilities of the manipulator and the geometric characteristics of the assembly parts.
ii. Quantify admissible uncertainties ranges for each developed method.
iii. Proposal of tools for the adaptation of the developed methods to general assembly tasks.
iv. Implementation of the developed methods on the mobile manipulator use case.

1.3. Project planning

This project was divided into several smaller tasks. Firstly, a literary review of robotic assembly stations and force control was performed since it was crucial for the understanding of the used tools and culmination on a flexible and robust solution. In parallel with this task, the used hardware and its functionalities were explored.

Subsequently, the gripper was integrated, establishing its communication with the manipulator's controller and programming its control functions, culminating on a fully functional setup. Then, all the different methods and strategies were developed to successfully accomplish the required assembly task. Finally, these methods were implemented in the mobile setup, validating the developed strategies.

Figure 1 displays a Gantt chart outlining the project organization throughout its timeline.

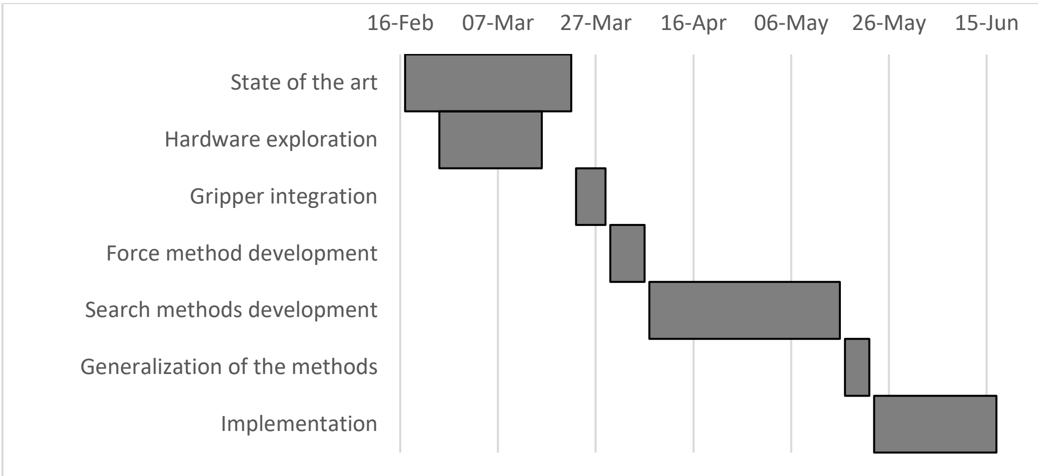


Figure 1. Gantt chart displaying the distribution of tasks

1.4. Dissertation structure

This dissertation was divided and organized in several chapters. Following the present introduction, the literary review will be exposed, where the state of the art of robotic assembly tasks and force control is introduced. Chapter 3 explains the studied topic in more detail and showcases the studied assembly task, exploring the degrees of freedom in which positional errors are present. Finally, this chapter lists all the hardware components used and details the inherent compliant tools of the used manipulator, which are the foundation on which all the compliant assembly methods will be built upon.

Subsequently, chapter 4 explains the integration of the gripper, including its communication solution and created control functions. It also introduces the setup used to develop the compliant assembly strategies and presents some considerations for the creation of said methods. Subsequently, it outlines all the established methods and strategies – contact force analysis method and search methods. Finally, a few tools and considerations are showcased, with the goal of expanding the scope of the present work to a generic method that can be implemented in any unstructured robotic assembly station.

Then, chapter 5 exposes the implementation of the developed methods on the actual mobile robot setup. Initially, it introduces the final setup of the mobile assembly station and established a correlation between the assembly workspace errors used during development and the mobile robot errors. This is followed by an exposition of the implemented methods and final quantification of the errors that can be overcome on both workspaces. Finally, the communication between the manipulator and mobile platform is exposed, as well as a brief explanation of the architecture of the mobile assembly station as a whole.

Finally, chapter 6 outlines the main conclusions of the present work and proposes topics that can be further explored in the future.

2. State of the art

This chapter is an outline of the literary research conducted for this dissertation. Firstly, an overview is given on the robotic assembly process, delineating the main characteristics required for its correct implementation. Subsequently, force control algorithms were introduced, giving emphasis on active force control. Finally, different robotic assembly strategies were outlined, assessing the advantages of each one.

2.1. Robotic assembly process

Originally, assembly tasks were predicted to be the dominant application for industrial robots. However, position uncertainties arising from unstructured work environments turned out to be a major hurdle for the implementation of robotic assembly. Thus, these operations are still mainly accomplished manually, as they require the dexterity only a human can offer [2], [7].

Manual assembly tasks are subject to typical shortcomings of human jobs, such as low efficiency and inconsistent end product quality. Furthermore, manual assembly of heavy components may cause repetitive stress injuries and ergonomic problems to the workers [8].

Recent breakthroughs in robotic force control, machine vision and robot dexterity may allow industrial robots to finally live up to the original vision of large-scale robotic assembly applications [1], [2].

Robotic assembly's success is highly dependent on the positional precision of the manipulator relative to its work environment. The use of highly sophisticated production lines, in which the position and orientation of the workpieces are known with high precision (structured work environment), usually leads to a simple implementation of assembly tasks. However, due to its high costs, it is only economically viable to implement them for mass production. On the other hand, discrete production, or production of small batch sizes, usually rely on unstructured work environments, which leads to higher position and orientation uncertainties. This is one of the biggest obstacles for the implementation of flexible robotic assembly stations, especially when the uncertainties are larger than the assembly tolerances between parts. Additionally, these uncertainties tend to lead to undesired contacts and collisions between workpieces and, therefore, external forces exerted on the manipulator [9]–[13].

Furthermore, due to the aforementioned uncertainties, a control strategy based solely on position control can be ineffective and insufficient for the successful implementation of an assembly task. Thus, it is necessary to implement a strategy capable of controlling not only the position of the manipulator, but also the force and torque profiles at its end-effector – implementation of compliance control [4], [9], [10].

Also, compliance control is an essential tool in robotic assembly as it limits contact forces, preventing damages to the workpieces. This is of particular importance when working with small and fragile workpieces [11]. This type of control is also crucial for collaborative robots (i.e. collaboration between robot and worker with a shared workspace) as it implements force control algorithms capable of minimizing unwanted collisions [2].

To sum up, in order to achieve comparable assembly performance to the equivalent manual task, or even surpass it, the robotic systems must have the following characteristics [9], [10]:

a) **Flexibility**, so that it can be easily adapted to multiple assembly tasks.
b) **Robustness**, to be able to adapt to changes without the need of human interference.
c) **Compliance**, to accommodate assembly tolerances smaller than the positional uncertainties and apply specific force and torque profiles needed for assembly tasks.

Only a system with these characteristics will be able to correctly interact with the environment and achieve a successful assembly task (e.g. meshing of gears, alignment and insertion of spline shafts, snapping of parts, etc.) [10].

2.2. Force Control

Force control algorithms increased the scope of robotics as they provided the essential tools for a robotic system to be able to apply a necessary force or to react to an external force in a predetermined manner. Furthermore, these algorithms may also be used as a way of exploring the work environment, reducing its uncertainties (e.g. evaluating the position and orientation of a component). Hence, the use of force control allows the implementation of robotics in new applications, such as surface finishing operations and assembly tasks [14]–[16].

Force control can be implemented actively or passively, depending on whether the force and torque measurements are fed back into the controller.

2.2.1. Passive force control

Passive force control is an open loop control algorithm without feedback of force information and, therefore, its force output is constant (cannot react to external stimuli). Although these methods usually have low accuracy and robustness, they do not require transducers, which leads to lower implementation costs when compared to active force control algorithms [14], [17]–[19].

The implementation of passive force control is usually accomplished at tool level (i.e. the robotic system controls its position and the tool is responsible for applying a constant force) [2], [17]. These tools can be used to accomplish simple assembly tasks with small position uncertainties and considerable assembly tolerances [1], [19].

Passive force control tools can employ different technological solutions, such as spring-controlled systems, mass counterbalanced systems and pneumatic actuators. Spring-controlled and mass counterbalance systems are simple to integrate but have limited applications. Pneumatic force control systems are more difficult to integrate but have other advantages that make them the most common technological solution for passive control tools – they are not affected by inertial loading, are easy to control and can have pressure regulators to easily adjust the force output) [2], [17].

2.2.2. Active force control

Active force control is a closed loop control algorithm with force measurement feedback. These algorithms implement a reactive control strategy that is accomplished at the controller level and allow the system to achieve a compliant behavior (i.e. the robotic system controls its position and force output directly) [2], [14], [18].

The implementation of active force control is usually more costly than passive solutions. However, these methods usually result in a more robust system, with better accuracy and repeatability. Furthermore, active and passive force control methods can be used simultaneously to achieve better cycle times and the ability to adapt to external stimuli [2], [17], [19].

The measurement of the external force acting on the end-effector can be obtained directly through a six-axis force/torque sensor. However, this sensor is costly and, therefore, other methods can be used to estimate the values of force and torque at the end-effector (usually at the cost of measurement accuracy). One of the best methods for estimating external force consists of the measurement of individual torque values at every joint and the use of the robot’s Jacobian for the calculation of external forces. Another alternative, for cases where there are no torque sensors at the joints, is to use a measurement of each motor’s current to estimate its torque. This method is less viable as it has a lot of variables that can influence its results, such as gravity and friction disturbance forces, especially when its transmission system uses gears. Lastly, external force estimation can also be achieved by an observer in the robots control system. This method is not commonly used as it requires a satisfactory dynamic model of the robot, which might be hard to achieve in some practical cases [12].

The following algorithms are the most relevant types of active force control for robotic assembly tasks:

Hybrid force/position control

Hybrid force/position control uses closed loop algorithms that calculate force and position error from the difference between its measured and desired values. The calculated force error is used in a control law to change the manipulator’s position and, consequently, tune its force output. Typically, a P, PI or PD controller are used in the establishment of the control law due to their inherent simplicity and effectiveness [2], [14], [20].

Due to the nature of this strategy, both control algorithms must switch between one another to keep position and force errors as low as possible. However, since force errors are corrected by adjusting the end-effector’s position, this control strategy may result in significant trajectory deviations. Furthermore, since robotic systems tend to aim for a high structural stiffness to achieve high positional accuracy, a small position deviation may lead to a high force variation on the end-effector. On the one hand, this may be beneficial for a hybrid force/position control strategy as it leads to smaller trajectory deviations, however, it can make the algorithm’s calibration more challenging [2], [14].

This control strategy usually leads to better results in low speed applications, due to increased reaction time tolerance. This increased reaction time has close to no influence at a control level but has enormous influence at a physical level, due to its limited physical response time (which is determined by its acceleration capacity – the ratio between motor torque and system inertia) [14].

Impedance control

Mechanical impedance measures how much a given system resists motion when subjected to a force, i.e. the ratio between the force exerted on a system's point and its resulting velocity.

A control strategy based on impedance aims to control the robotic systems contact response through the implementation of a dynamic relationship between the system's position and its applied force. Instead of controlling force and position with different algorithms, this strategy uses the system's position error to tune its force response. Thus, with impedance control, it is possible to control multiple variables with only one algorithm [6], [9], [18], [21], [22].

Typically, a second order linear differential equation is used as the dynamic relationship between force and position due to its simplicity and ease of control. This equation can be determined through the analogy of the chosen system variables with a typical mass-spring-damper. Hence, the dynamic relationship can be parametrized by two constants – stiffness and damping. These constants are responsible for the dynamic response of the system in situations of contact with external objects [2], [3], [6].

This control strategy provides a high bandwidth force response for situations of tool contact with rigid objects. Impedance control is the most developed and stable method for collision control and, therefore, it is an essential tool for force controlled assemblies [2], [18].

Admittance control

Mechanical admittance is the inverse of impedance, i.e. the ratio between the velocity of a system's point and its resulting resistive force. Similarly, admittance control is the inverse of impedance control – the system measures the force exerted on a given point and moves it with a certain velocity. Once again, the system's position and velocity are described in a single dynamic relation parametrized by stiffness and damping constants. This equation dictates the system velocity response to an external force [2].

With the use of this strategy, contact forces become dependent on the commanded velocity reference, which entails increased process cycle times for assembly tasks with limited collision requirements [9].

2.3. Assembly Strategies

Numerous different assembly strategies can be used to perform a given task, depending on the sensor information available. The simplest implementation uses pure position control of the robot and, as a result, is highly reliant on the accuracy of the robot and the work cell. Therefore, this strategy cannot handle significant part variation and positional inaccuracies and, thus, is inadequate for assembly operations in unstructured work environments. In fact, the use of this control strategy may require dedicated part feeders to avoid position uncertainties. Another shortcoming of using only position control is the difficult assessment of task success due to the lack of feedback during the assembling process [2], [12], [23].

It is possible to implement a position control strategy capable of overcoming some position uncertainties through the use of compliant tools and part guidance (e.g. with chamfers instead of sharp edges). For simple assembly tasks, this may be the preferred method as it usually leads to cheaper implementation costs when compared with more complex strategies [2], [9]. However, this solution may not be viable as the chamfer can compromise the working capacity of the part (e.g. in the insertion of a piston in the corresponding cylinder of an internal combustion engine, the presence of lead chamfers on the piston skirt and cylinder bore may compromise emission control and engine efficiency) [1], [10].

An alternative strategy for cases where position control is not sufficient, is the implementation of adaptive techniques using force control in event-based search strategies. This strategy can be used to surpass bigger uncertainties in the parts and work cell. The parametrization of the events is crucial as they trigger transitions between assembly steps. For instance, a typical peg-in-hole assembly may be accomplished using a blind search method around the candidate hole position in order to find its exact position. In this case, the peg may be moved either randomly or by a structured geometrical pattern (spiral, circle, sinusoidal wave or linear). The search motion continues until a certain force and torque condition, meaning that the peg is aligned with the hole, is triggered and the next step can begin. This assembly can also be accomplished through smarter methods that probe the candidate hole position and evaluate the force and torque sensory data to identify the location and orientation of the hole [2], [7], [8], [12], [13].

Vision based strategies are another alternative for assembly tasks. In this approach, a computer system examines images captured by a camera and outputs the manipulator configuration (6 degrees of freedom) needed to accomplish a given assembly task. When correctly implemented, this method can react to much higher uncertainties, tremendously increasing process flexibility. However, the performance of assembly processes based on vision are highly reliant on present light conditions and calibration between the vision system and the robot's workplace (i.e. the mathematical relationship between workspace coordinates and pixels in digital images). Moreover, the calibration process is typically troublesome and is often responsible for the unsuccess of this method [1], [23].

A hybrid method based on vision and force control may be the best bet for complex assembly operations. In this method, the vision system is responsible for an initial analysis of the workspace and plotting of relevant points. This information is then used by an event-based assembly strategy using force control in order to complete the assembly. The combination of these two strategies leads to a more flexible and robust process, as the initial vision analysis allows the system to deal with a high degree of uncertainties and the use of force control decreases the need for a rigorous calibration of the vision system [1].

3. Topic and hardware description

This chapter provides a more comprehensive explanation of the proposed project, as well as the used hardware. Firstly, the topic at hand and its connection to the ScalABLE 4.0 project will be described. Afterwards, the assembly task used to test the effectiveness of the system will be showcased, exploring the degrees of freedom in which the aforementioned uncertainties are present. Additionally, every hardware component of the setup will be specified, including the required interactions between them. Finally, the inherent compliant capabilities of the manipulator will be thoroughly explored, as they are the essential tools for the development of a mobile assembly station using force control to overcome position uncertainties.

3.1. Project Description

The studied system consists of a manipulator transported by a mobile robotic platform developed in the scope of the ScalABLE 4.0 project. This system was designed to operate as a mobile assembly station, with the capability of moving between workstations, transporting assembly parts between them, and performing the required assembly tasks.

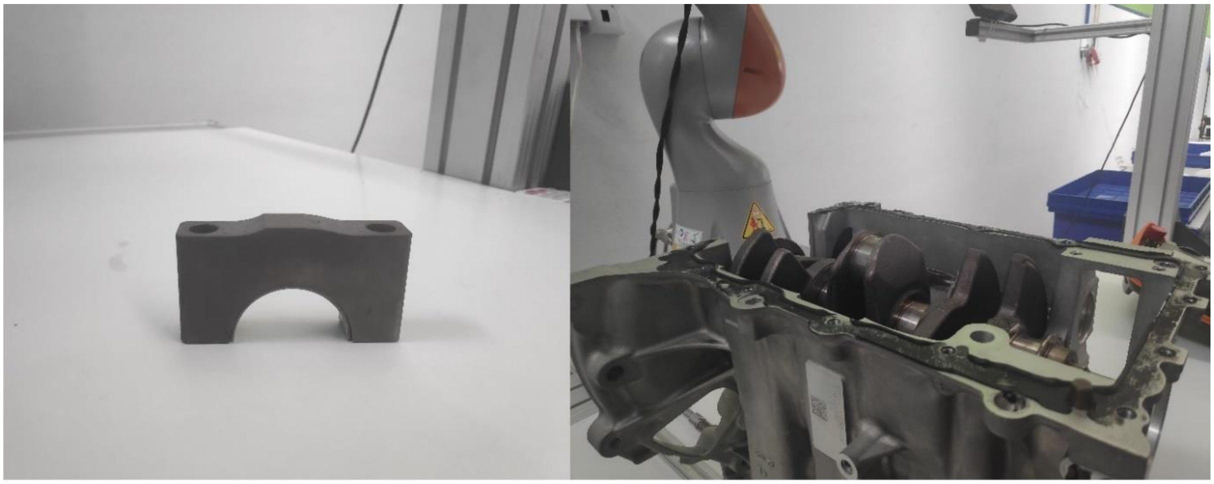
However, the described mobile robot will always be responsible for introducing some degree of positional uncertainties to the system, as its location tracking and positioning system cannot ensure the needed precision to easily accomplish the assembly task. Even if the assembly is done in a highly structured workspace with close to no uncertainties, the spatial errors associated with the positioning of the mobile robot will always introduce positional errors to the assembly task.

The literature reviewed in the previous chapter confirms that positional uncertainties are, in fact, one of the biggest challenges for the implementation of large-scale robotic assembly stations in unstructured work environments. Further, it suggests that the best solutions for overcoming these uncertainties are, either through a computer vision system or with the implementation of compliant behavior in the manipulator.

The present dissertation aims to explore the capabilities of force control as a tool to overcome the positional uncertainties created by the mobile transportation robot, culminating in successful assembly tasks.

3.2. Assembly task description

The proposed assembly task is the positioning of a main bearing cap on a rod bearing journal of a crankshaft (Figure 2). This operation assumes a previous assembly of the crankshaft on the engine block and main bearing on the bearing cap. While this assembly has no inherent significance to the working of the engine, it shows great potential for this study, as it is isolated enough so that big uncertainty values can be tested without collision with other parts, thus removing further elements that might influence the obtained results.



a) Main bearing cap b) Crankshaft mounted on the engine block

Figure 2. Components of the studied assembly task

The reference frame displayed on Figure 3 will serve as a basis for all further discussions. As such, if no other indication is given, a discussed axis will always refer to this frame.

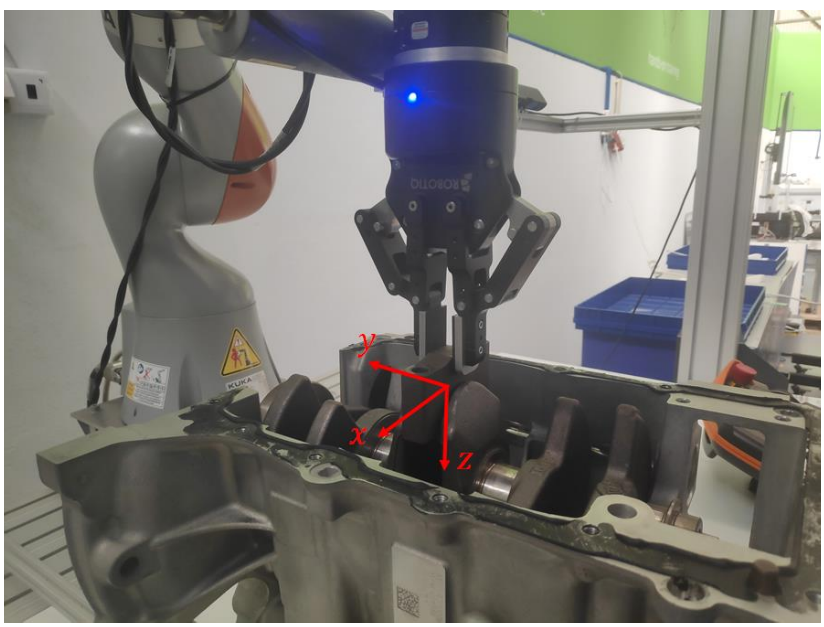


Figure 3. Main reference frame of the studied assembly task

As it was previously discussed, all the positional uncertainties present on this assembly task arise from positional errors of the mobile robot, which moves within the $_xO_y$ plane. As such, the positional errors mentioned only occur on this plane and, thus, the assembly uncertainties present are the following:

- Translation along the x axis.
- Translation along the y axis.
- Rotation around the z axis.

There are no errors on the remaining degrees of freedom as the robot is mounted on top of the mobile robot and the engine block is located on top of a fixed table, both with constant heights and orientations.

3.3. Hardware specifications

The aforementioned mobile assembly system is composed of two main components: the mobile robot and the robotic manipulator mounted on it. The former was developed within the frame of the ScalABLE 4.0 project and, therefore, will not be discussed in detail as it falls outside the scope of this study. Thus, this section is a detailed exposition of the robotic manipulator and its setup (basic hardware components, end-effector tool and software used for its programming).

Figure 4 displays the basic setup of this industrial robot; however, tools and other task reliant equipment may be necessary. The end-effector tool selected for this project was a compliant gripper capable of grasping and carrying the required workpieces in a force-controlled manner.

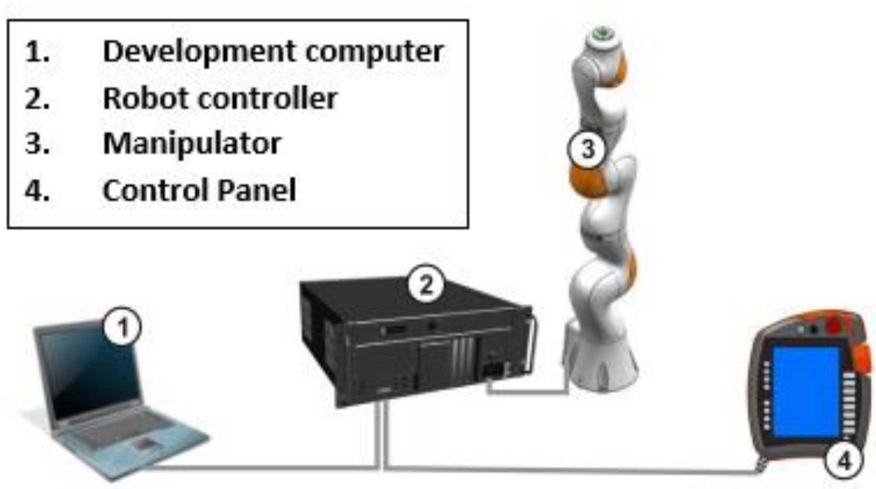


Figure 4. Basic setup components of the manipulator [24]

3.3.1. Manipulator

The manipulator used in this task was the KUKA LBR iiwa, a lightweight collaborative robot developed in 2013 by KUKA Industrial Robots, one of the world's leading robot manufacturers. It is a 7-axis robotic arm equipped with a joint-torque sensor in each axis, allowing it to achieve a compliant behavior – it is capable of detecting contact forces and reacting appropriately to them in a force-controlled manner.

This manipulator was one of the world's first series-produced collaborative robots, meaning that it may work alongside a human, safely sharing the same workspace on highly sensitive tasks (scrapping the need for physical separation and safety fencing between the robot and worker). With this approach, the robot does not replace the worker, but complements his capabilities and relieves him of arduous, repetitive tasks. Moreover, collaborative robots open the possibility for new robotic applications and tend to increase cost-effectiveness and resources efficiency.

KUKA recommends the use of the LBR iiwa for the following applications:

- Painting/gluing
- Palletizing
- Measuring/inspecting
- Machining
- Assembly
- Handling

The LBR iiwa's traits of dexterity, kinematic redundancy and inherent compliance are some of the features commonly present in the new generation of robotic manipulators. These attributes are particularly important for assembly tasks as they are essential for insertion, screwing and handling of delicate parts [25].

The version of this robotic manipulator used in the present work is the LBR iiwa 14R 820. Table 1 summarizes the main characteristics and data of this specific version.

Table 1. Main specifications of the KUKA LBR iiwa 14R 820 [26]

Rated Payload	$14\ kg$
Weight	$29.9\ kg$
Payload-weight ratio	0.47
Number of axis	7
Positioning accuracy (ISO 9283)	$\pm 0.1\ mm$
Positioning accuracy (ISO 9283)	$\pm 0.15\ mm$
Maximum reach	$820\ mm$

3.3.2. Robot controller

The robot controller supplied for this manipulator is the KUKA Sunrise Cabinet, a robust controller with an integrated control PC that can be used with every KUKA lightweight robot. This controller is responsible for the manipulator's safety configuration, motion control, logic setup and process control.

The Sunrise Cabinet is also equipped with batteries, allowing the controller to be shut down in a controlled manner in case of a power outage. It is equipped with several interfaces, for the connection of the manipulator, the control panel, the development computer and other devices (e.g. a fieldbus network).

This controller can be operated in several different control modes and has numerous functionalities for the implementation of compliant control. These tools are of major importance for this study and, thus, will be discussed further in Section 3.4.

3.3.3. Development Computer

Typically, the programming of an industrial robot can be done online, via its control panel, or offline using the appropriate software running on a development computer. KUKA´s LBR iiwa has the particularity of only being able to be programmed offline and, therefore, a dedicated development computer is an integral component of the setup.

The language used for programming this robot is Java and the main software needed for this task is *KUKA Sunrise.Workbench*, a modified version of the well-known Java Integrated Development Environment (IDE) – Eclipse. This software is used as the development environment for the robotic station and it offers all the functionalities of a typical IDE, essential for the development of robot applications. Moreover, it offers numerous Java packages and classes used for the configuration and operation of the robot, functionalities for the setup of safety configurations, remote debugging and project synchronization with the controller.

Another important software is *KUKA WorkVisual*, which is used mainly for bus configuration and variable mapping in situations of communication through a computer network protocol. This software automatically generates a Java class containing the methods required to directly access the mapped inputs or outputs. The fieldbus systems supported by Sunrise that can be configured with WorkVisual are PROFINET, PROFIBUS and EtherCAT.

3.3.4. Control Panel

The control panel supplied for the use with this manipulator is KUKA smartPAD, a hand-held touchpad that cannot be used for programming as it is usual practice in industrial robots. Nevertheless, it still maintains multiple functionalities such as managing robot and background applications, activating the safety configuration, creating and teaching frames, jogging and calibration.

Some applications may control the running robot tasks remotely, eliminating the need of managing them in the control panel. In these cases, after activating the safety configuration and calibrating the needed frames, the control panel may be disconnected from the controller.

3.3.5. Gripper

The tool used in this project was Robotiq's 2F-85, an adaptative gripper particularly suited for the use with collaborative robots as it can control its exerted force. Figure 5 displays the versatility of this tool, not only due to its capability to adapt itself to different shapes of the manipulated object, but also since it can grasp objects externally (with a closing motion) or internally (with an opening motion).

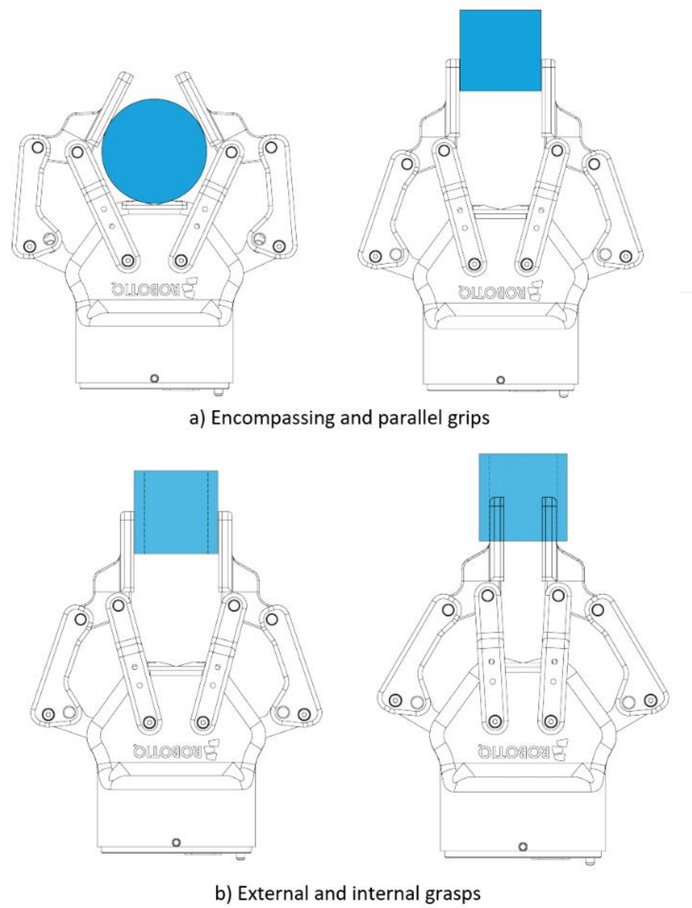


Figure 5. Versatility characteristics of the Robotiq's 2F-85 gripper [27]

Additionally, there are numerous fingertips options that can be interchange in order to better suit the given task.

Table 2 displays the most important working specifications of the Robotiq 2F-85. This set of data is important as it shows the working capabilities of the gripper.

Table 2. Working specifications of the used gripper [27]

Stroke	$85\ mm$
Grasp force	$20 - 235\ N$
Finger speed	$20 - 150\ \frac{mm}{s}$
Position repeatability	$0.05\ mm$
Position resolution	$0.4\ mm$
Force repeatability	$\pm 10\ \%$

As the tool is mounted on the manipulator's end-effector, it will affect its dynamic reaction. Thus, the mechanical specifications of the gripper need to be accounted by the controller, so that it can properly plan the programmed motions with the correct values of torque applied in each joint's motor. The most relevant mechanical characteristics are its weight, center of mass and moments of inertia, all of them provided by the supplier.

The communication with the gripper can be accomplished directly via Modbus Remote Terminal Unit (RTU) (a well-known and widely spread serial communication protocol), through a RS 485 signal. Figure 6 displays the wiring scheme of the used gripper.

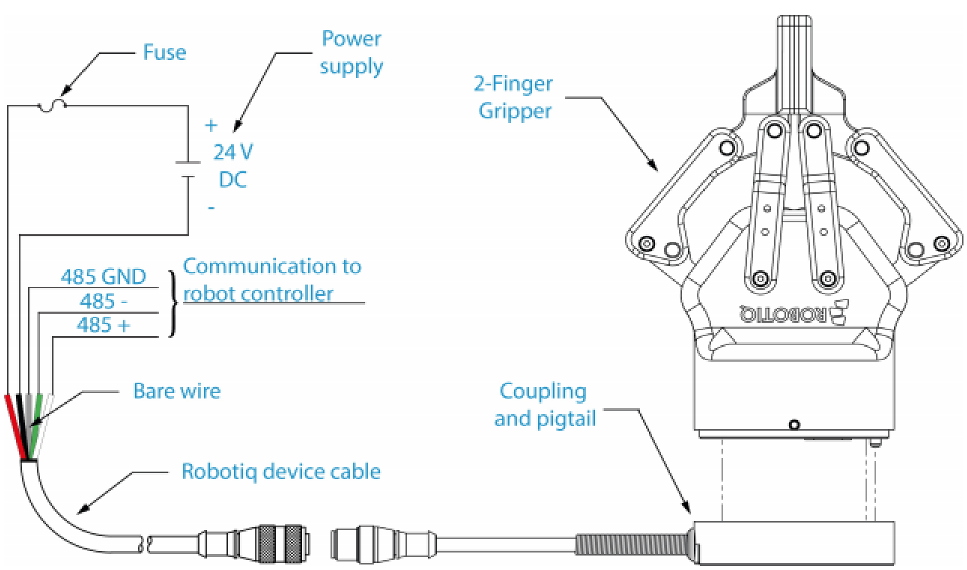


Figure 6. Gripper's wiring scheme [27]

However, this is not the ideal configuration for integration with the manipulator since, ideally, the gripper's control would be accomplished within the robot application running on the manipulator's controller. For this purpose, an optional hardware component was used – Robotiq Universal Controller.

This component does the conversion between the RS485 signal and an ethernet based communication protocol. The universal controller can work with the following protocols:

- Ethernet/IP
- Modbus TCP
- EtherCAT
- PROFINET
- DeviceNet
- CANopen
- Profibus

Thus, for the gripper to be integrated into the system, the communication between the manipulator's controller and Robotiq Universal Controller must be set up. This configuration and the relevant gripper control functions will be exposed in more detail during the solution development chapter, specifically in section 4.1.

3.4. KUKA LBR iiwa compliant control

The used manipulator is equipped with a torque sensor in every axis, which enables it to estimate the external force applied on a certain point. Hence, it has an inherent capability of implementing compliant behavior without the need of an external force/torque sensor. This section outlines all the inherent control modes available for the manipulator, as well as other relevant compliant tools.

3.4.1. Available control modes

The robot's controller is equipped with multiple different control modes, each with its inherent characteristics and applications. There are four different servo controllers available:

Position Controller

This control mode is the classic position control used in most industrial robots. It aims to execute the specified path with the maximum positional accuracy. It does not use the equipped torque sensors and, thus, external influences such as obstacles or forces are not accounted for.

Cartesian Impedance Controller

As it was mentioned in the presented literature review, this control mode is particularly relevant in the scope of assembly tasks. Under impedance control, the system is modeled on a virtual spring damper system, where the values of stiffness and damping can be configured.

Using this control mode, the exerted force by the manipulator at the contact point is reliant on its positional deflection and the configured stiffness value (Figure 7). This correlation can be estimated using Hooke's law (Equation 1).

$$F = C \cdot \Delta x \tag{1}$$

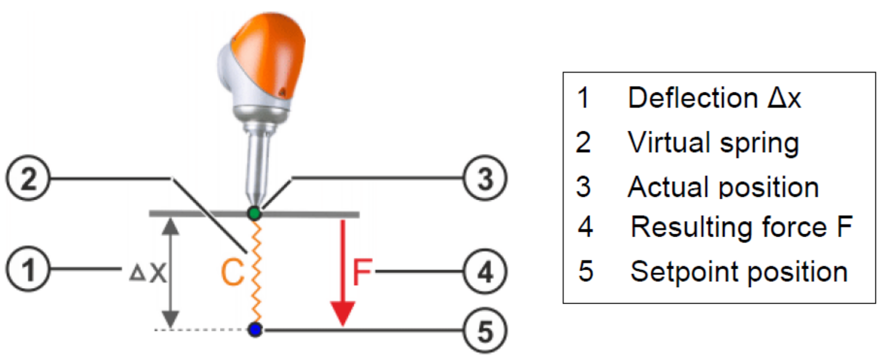


Figure 7. Cartesian impedance control exposition [24]

When controlled by the cartesian impedance controller, the manipulator assumes a compliant behavior, in which its dynamic response can be adjusted accordingly to the needs of the task (e.g. restriction of contact forces or adjusting the position deflection caused by a certain external force or obstacle).

In addition to the values of stiffness and damping, multiple other parameters can be configured. These parameters may limit the maximum force value exerted by the manipulator, its maximum path deviation, its null space stiffness (the stiffness of the redundancy degree of freedom), among others.

Cartesian impedance controller with overlaid force oscillation

This control mode is an extension of the aforementioned cartesian impedance controller, as it has the same basic behavior and is configured using the same parameters. However, in addition to the compliant behavior, this mode allows the introduction of constant force overlays or sinusoidal force oscillations. As the system is under impedance control, these force overlays result in path deviations.

Figure 8 illustrates the path deviation caused by an overlaid force oscillation in the x axis, while the manipulator moves along the y axis.

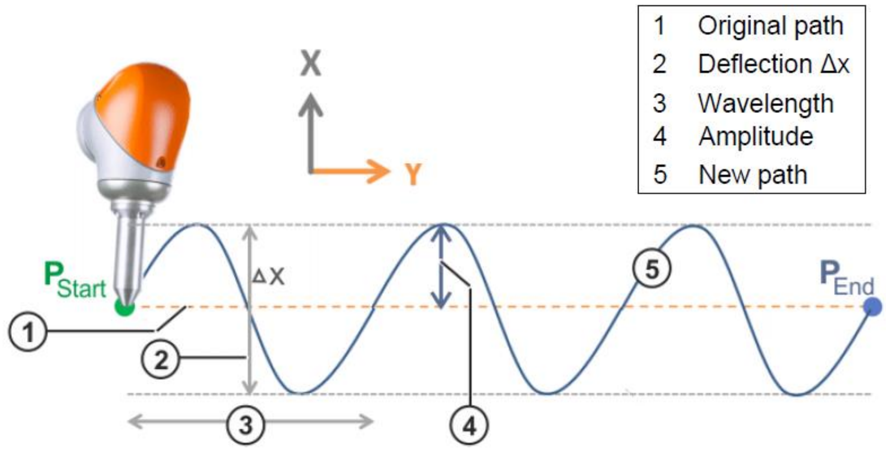


Figure 8. Example of the cartesian impedance controller with overlaid force oscillation [24]

For the configuration of this control mode, in addition to the impedance control parameters, it is necessary to configure the magnitude of the force oscillation, its frequency and the degree of freedom in which it will be applied. Once again, the amplitude of the deflection can be predicted by the magnitude of the force and the configured stiffness value, according to Hooke's law.

Additionally, it is possible to combine multiple force overlays to form other oscillation patterns (e.g. sinusoidal path, Lissajous curve, spiral).

The cartesian impedance control with overlaid force oscillation may be relevant in the scope of this project as a way of implementing blind search motions, while limiting collision forces between workpieces.

Axis-specific impedance controller

This control mode is a different implementation of the impedance control mode, where the configurable stiffness and damping values are set individually for each joint, instead of being parametrized for the cartesian degrees of freedom of a given reference frame.

3.4.2. Other compliant tools

In addition to the described control modes, which allow the implementation of compliant motions, the signals of the joint's torque sensors can be accessed at any time, even when using the position control mode. These signals can be used to set up logic conditions to start/stop a given motion or even to perform different paths depending on the force condition sensed at a given moment.

It is important to note the cartesian force measurements used are just an estimation based on the values of the torque measured at each joint. As such, these values are not as reliable and accurate as the values obtained using a dedicated force/torque sensor on the tool. They are particularly prone to errors near singularity points, when some joints are fully extended, which causes unreliable torque measurements.

4. Solution development

This chapter exposes the proposed assembly routines developed in this study. Firstly, the gripper was integrated with the manipulator so that they could work as a cohesive unit. Subsequently, the development setup will be introduced, emphasizing how the error was quantified in the assembly workspace, as well as how the present errors impact the performed motions. Then, the developed methods will be exposed – contact force analysis and search methods. Finally, a few tools allowing the developed solutions to be implemented in general assembly tasks will be proposed.

4.1. Gripper integration

As it was mentioned in Subsection 3.3.5., in order to control the gripper from within the manipulator's controller, an extra component can be used – the Robotiq Universal Controller. Thus, to fully integrate the gripper with the manipulator, the communication with the Robotiq Universal Controller must be configured and the gripper control functions must be programmed.

4.1.1. Communication setup

Within the current application, the communication protocol used between the manipulator's controller and the Robotiq Universal Controller was EtherCAT, as it is suitable for real-time computing requirements and is supported by the KUKA Sunrise Cabinet. Figure 9 displays the communication scheme between the gripper, the Universal Controller and the robot controller.

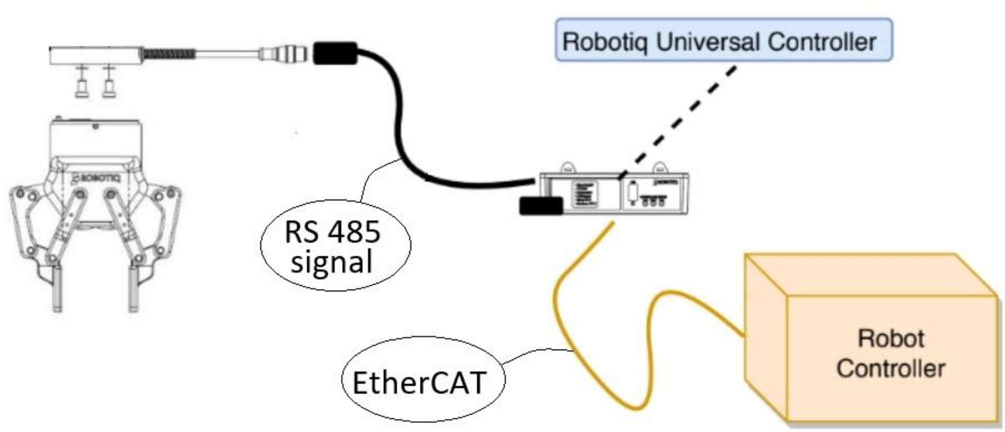


Figure 9. Gripper communication scheme [27]

The configuration of the EtherCAT network was performed automatically in the software KUKA WorkVisual using a "Universal Controller Configuration" XML file supplied by the gripper's supplier (Figure 10).

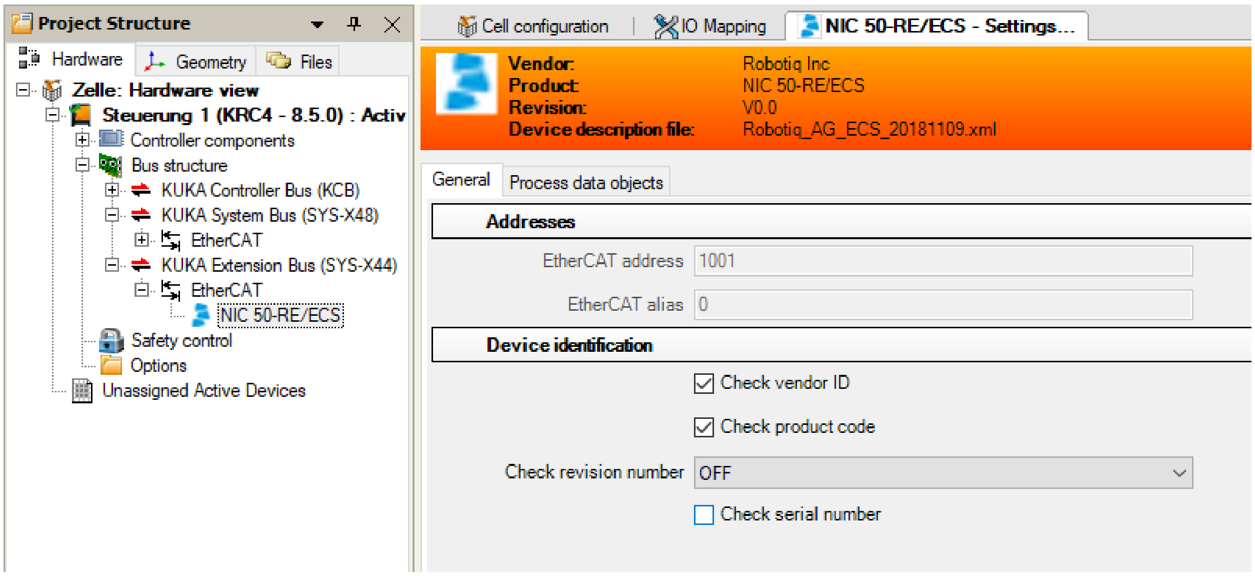


Figure 10. EtherCAT configuration for Robotiq Universal Controller generated in KUKA WorkVisual

Table 3 lists all gripper's control registers – 14 variables in total (functionality variables correspond to orders given to the gripper, whereas status registers correspond to the current gripper status).

Table 3. Gripper control registers [27]

Register	Robot Output / Gripper Functionalities	Robot Input / Gripper Status
Byte 0	Action Request	Gripper Status
Byte 1	Reserved	Reserved
Byte 2	Reserved	Fault Status
Byte 3	Position Request	Pos Request Echo
Byte 4	Speed	Position
Byte 5	Force	Current
Byte 6	Reserved	Reserved

These registers were automatically created in WorkVisual by the aforementioned XML file. Using this software, it is possible to create corresponding variables in KUKA Sunrise IO and map them accordingly, as seen in Figure 11.

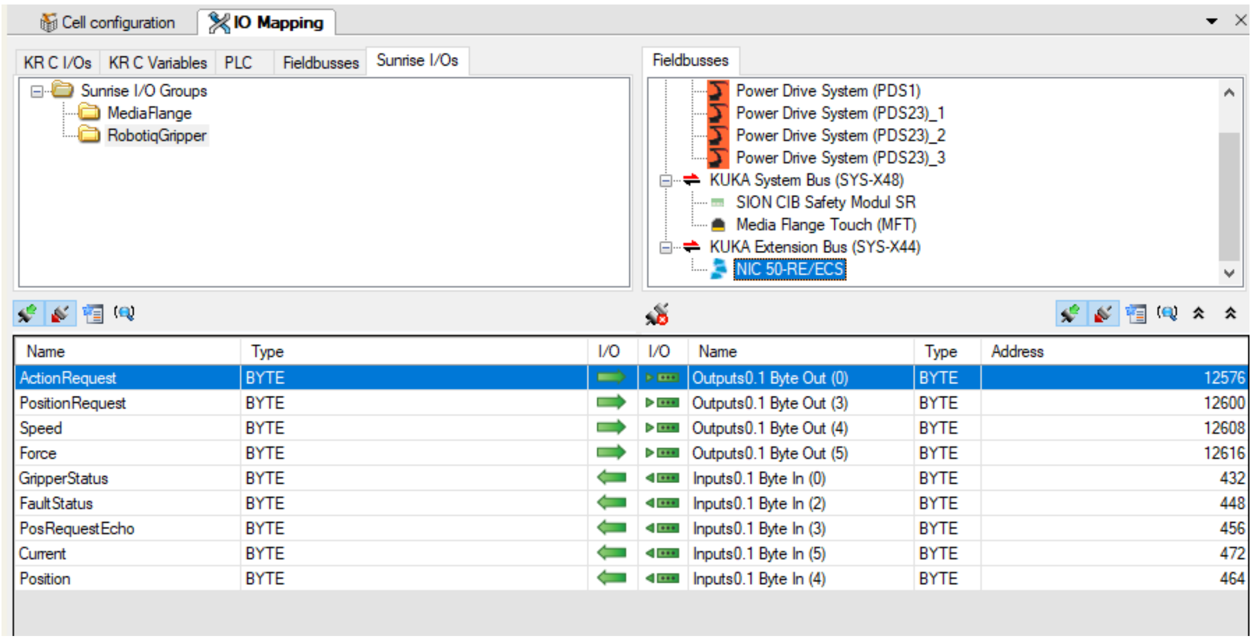


Figure 11. Variable mapping between Robotiq Universal Controller and KUKA Sunrise IO

The IO configuration created can then be exported to KUKA Sunrise WorkBench, where it automatically generates a Java class ("RobotiqGripperIOGroup.java") containing methods to write gripper´s functionality records and read its status records.

4.1.2. Control functions

With the above-mentioned configuration, it is possible to access the gripper control registers from within applications running in the manipulator's controller. Since these registers correspond to the communication variables with the gripper, they are essential for creating the required control functions. With this goal, a Java class extending the automatically generated "RobotiqGripperIOGroup.java" was created (being a child class, it inherits its methods for reading and writing gripper records). In this class, all the basic control functions were programmed.

This section is an exposition of all the created control functions and a brief explanation of their purpose. For further understanding on the individual bit manipulation of the control registers, the gripper's user manual, [27], should be consulted.

1. activate ():
    - Activates the gripper and waits for activation completion.
2. deactivate():
    - Deactivates the gripper
3. goToPosition(position, speed, force):
    - Moves the gripper to a certain position, with a given speed and force.
    - The admissible inputs have no physical meaning and their admissible range is $[0, 255]$.
    - Position = 0 corresponds to a fully opened gripper and Position = 255 corresponds to the fully closed position.
    - For the speed and force inputs, the minimum possible value corresponds to 0 and the maximum value corresponds to 255.
    - The function checks if the given inputs lie within the admissible range. If they do not, the function returns an error.
4. goToWidth(position, speed, force):
    - Moves the gripper to a certain position, with a given speed and force.
    - The admissible inputs have physical meaning.
    - Position represents the opening width - $[0, 85]\ mm$.
    - Speed represents the opening or closing linear velocity of the gripper's fingers - $[20, 150]\ mm/s$.
    - Force represents the maximum force exerted by the gripper - $[20, 235]\ N$.
    - The function checks if the given inputs lie within the admissible range. If they do not, the function returns an error.

4.2. Development setup and error simulation

The methods and algorithms described in this chapter were developed with the ultimate goal of being implemented in the mobile assembly station, working on an unstructured work environment. However, for development and testing purposes, the use of the mobile robot would not be viable since the positional errors associated with it would not allow accurate testing. This is a natural assumption because if the robot was accurate enough to position itself to a specific error value to be tested, it would be accurate enough to position itself with no error, thus eliminating the need of implementing compliant routines to overcome said errors.

Hence, an alternative setup was created with both the manipulator and the engine block mounted on a fixed location (Figure 12). Since the position of the rod bearing journal of the crankshaft is constant in relation to the manipulator, this setup eliminates all positional uncertainties of the assembly task (implementation of a structured work environment).

Figure 12. Development setup

The use of this configuration allows the successful completion of the assembly task using exclusively positional control. Likewise, it is possible to calibrate an approximation position to the assembly task, i.e. setting the bearing cap in a position where the assembly task can be completed by performing a vertical motion. Figure 13 demonstrates the approximation position for the studied task.

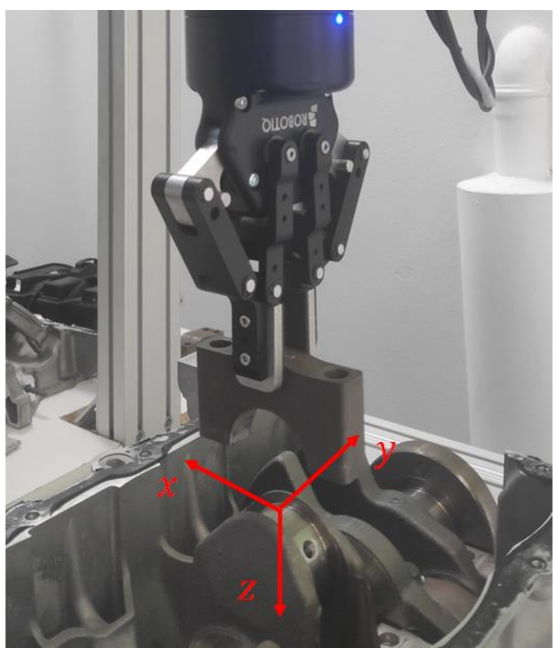


Figure 13. Approximation position of the studies assembly task

Similarly, it is possible to offset this position in the $_xO_y$ plane, moving the bearing cap with the desired positional and orientation error relative to the perfect approximation position. This method can be used to impose precise errors on the assembly workspace (x_A, y_A and α_A), which is imperative for the implementation of consistent and accurate testing.

Another important factor that should be accounted is the reference frame relative to which the manipulator's movements are programmed, as well as the impact caused on it by orientation errors. Also, the created method for simulating uncertainties differs from the real application of the mobile robot – on the former, the manipulator's base is placed on a fixed location, whereas on the latter, it is mounted on the mobile robot and, therefore, is also affected by the rotational error around the z axis.

As such, it is possible to infer that the errors of the real application with the mobile robot act on a manipulator level (all the possible reference frames of the robot are affected by the orientation error), while the errors introduced by the simulation method of the development setup only affect the reference frames located hierarchically bellow the robot flange (since the base of the manipulator has a fixed orientation relative to the engine block).

Therefore, a search routine developed and tested on the fixed setup can only be implemented in the real application if all its movements are programmed relative to either the reference frame of the manipulator flange or some frame hierarchically bellow it. As such, all the proposed methods on this chapter use the gripper's tool center point frame as a reference for all its motions, ensuring the validity of the analogy between both setups.

Moreover, all the stated error values for the methods and strategies developed in this chapter are quantified in the assembly workspace, A. These error values do not directly correspond to the positional errors of the mobile robot (x_R, y_R and α_R) but are influenced by them – this relationship will be further outlined during the implementation chapter, more specifically in section 5.1.

4.3. Contact force analysis method

In assembly tasks, it is expected that different positional errors relative to the approximation position lead to different interactions with the environment. Therefore, it is possible to acquire information about the positional errors by analyzing the measured contact forces and torques.

The relationship between the contact forces and the positional errors arising from workspace uncertainties are reliant on the parts geometries and their relative motion. Therefore, it is clear that there is no general rule for establishing this relationship as it is highly reliant on the task at hand.

Figure 14 displays two different assembly instances (each with a specific error relative to the defined perfect approximation position), as well as the expected contact forces and torques generated when a controlled descending motion creates a collision between the bearing cap and the crankshaft. The red and green reference frames represent the previously predetermined assembly reference frame and the "Tool center point frame", respectively. The blue vectors are a visual representation of the expected contact force and resulting torque for each contact position.

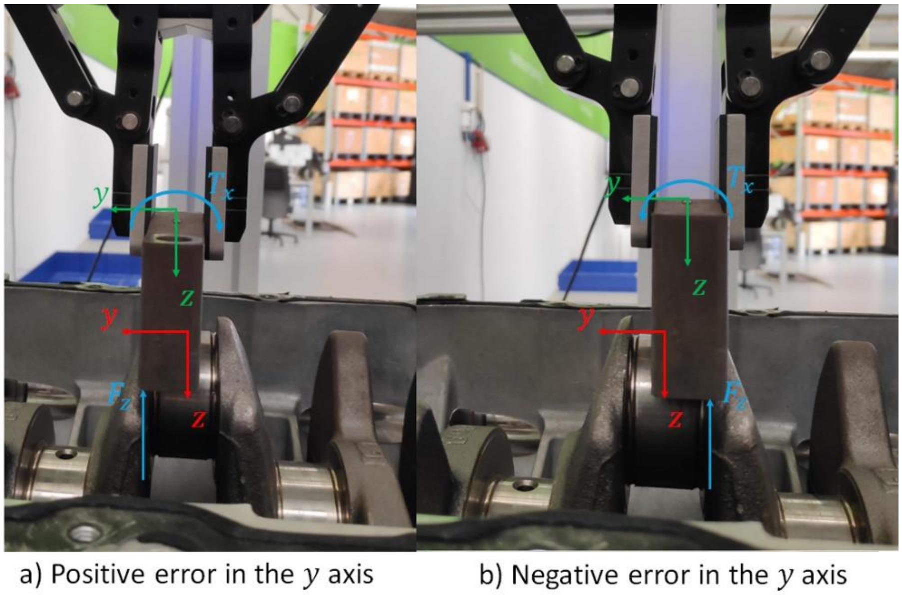


Figure 14. Expected forces and torques resulting from different error values

As it is displayed in Figure 14, a positive translation error in the y axis results in an expected negative torque in the direction of the x axis, whereas a negative translation error in the y axis has the opposite expected reaction (positive torque in the x axis).

Furthermore, bigger errors in the y axis lead to a smaller distance between the contact point and the center of the bearing cap, which ultimately results in smaller expected torque values. Since the magnitude of the measured torque is an indication of the error value, this method can be used as a way of accessing and overcoming positional errors.

In order to test the feasibility of this method, a routine was developed where the manipulator performs a vertical descending motion (positive relative to the z axis). This motion is aborted when it measures a vertical force of a given magnitude – detection of collision condition. In order to analyze the contact forces and torques, an inherent data recorder of the LBR iiwa was used (implemented with the Java class DataRecorder, in the roboticsAPI package). With this class, it is possible to sample the manipulator's position, force or torque measurements relative to a specified frame and store them in a *.log* file.

To analyze this data, it is important to understand that the values are not obtained directly from a force and torque sensor coupled to the tool but instead are estimated through the torque sensors in each axis of the manipulator. The estimated values would only be completely accurate when using a perfect model of the manipulator, tool and workpieces, which is hard to implement in real applications. As such, it is important to keep in mind that these values are subject to signal noise, making it hard to detect and evaluate low force and torque collisions. This shortcoming is particularly relevant in the implementation of low contact forces assembly tasks.

In order to compare the measured values with the expected results, the described routine was used to test the contact forces and torques for the error values displayed in Figure 14 (positive and negative translation relative to the y axis). For these tests, a collision detection force of 25 N was used.

The following graphs (Figure 15 and 16) display the evolution of the force and torque values measured relative to the “Tool center point frame” for the different error instances.

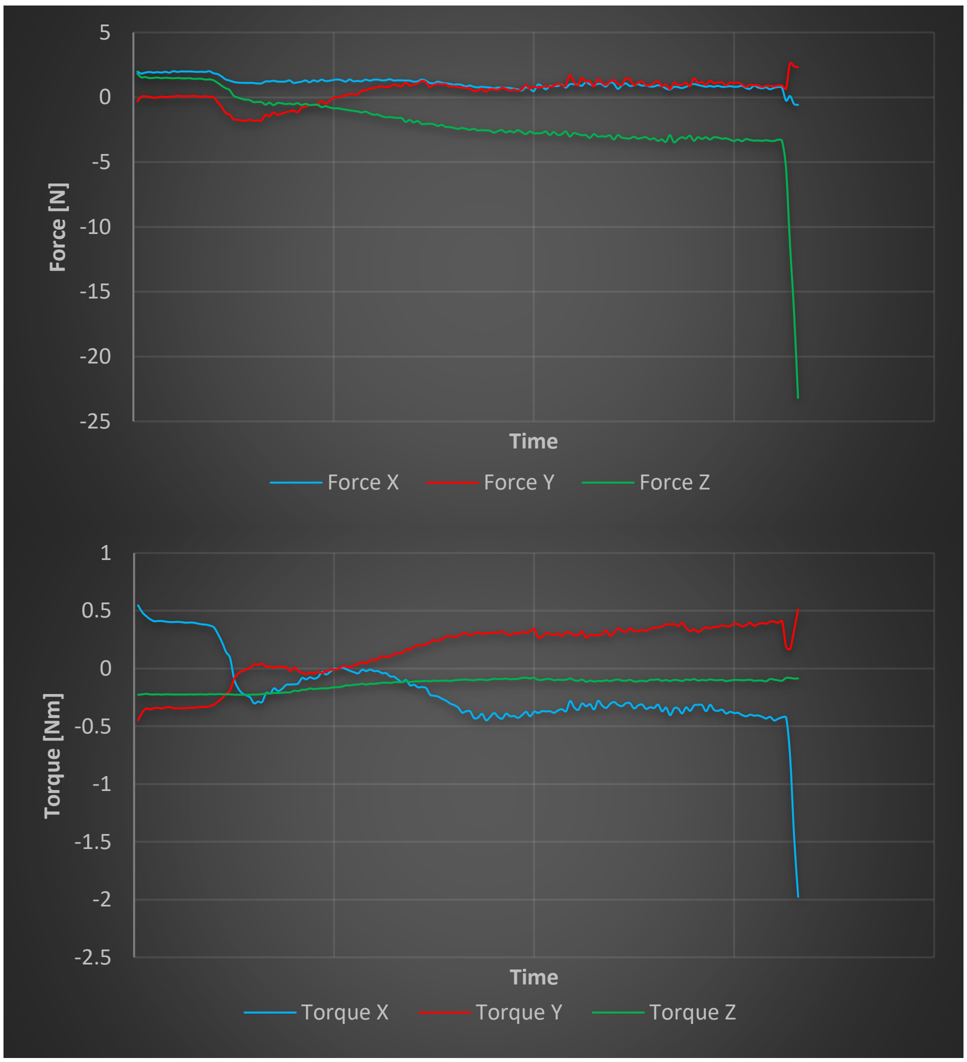


Figure 15.Force and torque evolution graphs for positive errors along the y axis

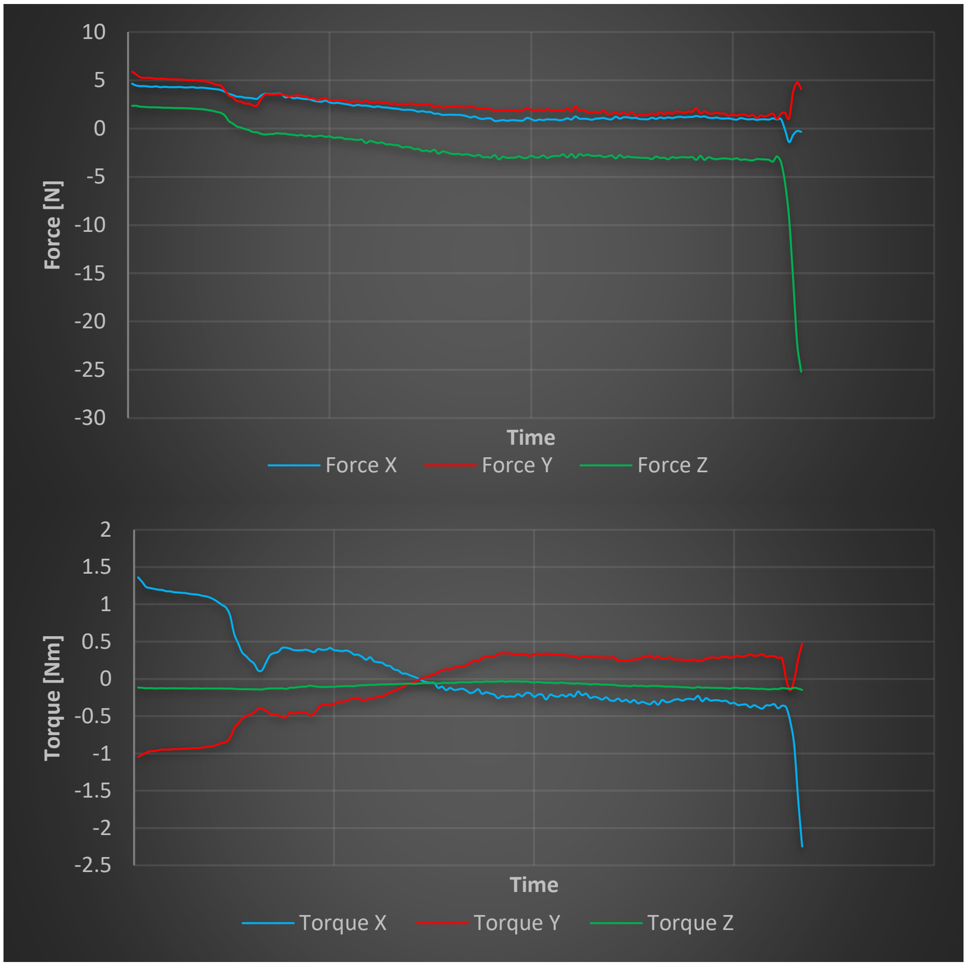


Figure 16. Force and torque evolution graph for negative errors along the y axis

Table 4 displays a comparison between the expected values and the measured data recorded in these tests.

Table 4. Comparison of the expected and measured force and torque values

	Force		Torque	
	Expected	**Measured**	**Expected**	**Measured**
Positive y error	Negative in the z axis	Negative in the z axis	Negative in the x axis	Negative in the x axis
Negative y error	Negative in the z axis	Negative in the z axis	Positive in the x axis	Negative in the x axis

Even though the predictions were corroborated by the measured forces, they were contradicted by the measured torques. Furthermore, since the measured torque response was indistinguishable for both situations, it is not possible to use this contact measurements to infer about the positional error relative to the y axis.

In fact, since the width of the bearing cap is of approximately $18\ mm$, the maximum distance between the contact force and the center of the bearing cap is $9\ mm$. This geometric characteristic of the studied assembly task entails that even when using a moderate force condition of $25\ N$, the maximum observed torque will be approximately $0.225\ N \cdot m$ (as demonstrated using Equation 2).

$$T = F \cdot d \tag{2}$$

$$T = 25\ N\ \cdot 9\ mm\ \cdot\ 10^{-3}\frac{m}{mm} = \ 0.225\ N \cdot m$$

The measured torque graphs displayed above show that the signal noise, caused by the estimation of the torque value through the joint torque sensors, leads to measurement uncertainties of approximately $\pm 0.5\ N \cdot m$. Hence, this measurement method is not adequate for the studied task as it is not viable to detect a torque with a maximum value of $0.225\ N \cdot m$ with a measurement method with uncertainties of $\pm 0.5\ N \cdot m$.

Thus, this method of overcoming positional errors through the analysis of contact force and torque cannot be implemented for the studied assembly task, when using this method of estimating the contact forces. However, this is still a relevant technique for generic assembly tasks as it is feasible to suppose that this approach could be implemented either by using a dedicated force and torque sensor (capable of more accurately detecting small torques), or by performing a different assembly task, in which the inherent geometric characteristics would translate in higher distances between the contact positions and the measurement points, resulting in higher torque values for low contact forces.

4.4. Search methods

This section serves as an exposition of the developed compliant strategies created to overcome positional and orientation errors. These strategies implement the inherent compliant control modes and tools of the KUKA LBR iiwa (detailed in Section 3.4.) and serve as instruments for the development of compliant search methods for the accomplishment of the studied assembly task.

4.4.1. Developed compliant search strategies

Direct impedance control convergence

Some assembly tasks are inherently convergent, meaning that the contact forces arising from the interaction between the components automatically correct the workpiece's position, rectifying the initial error of, at least, one degree of freedom. As such, the use of impedance control exploits this characteristic, as it allows the manipulator to move away from its planned trajectory, correcting the errors of the convergent degrees of freedom.

The studied assembly task has a cylindrical contact surface and, thus, is convergent for relatively small translational errors along the x axis. As such, the implementation of an impedance controlled vertical motion ought to be enough to overcome positional errors along this degree of freedom. This motion should be configured with a small stiffness value relative to the x axis to ensure position correction with low contact forces. Figure 17 demonstrates the contact force present for the described case - F_x is the force component responsible for the mentioned convergence.

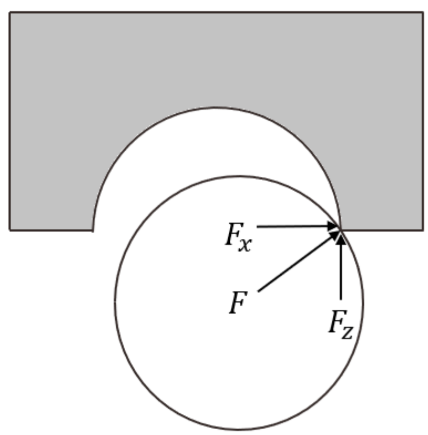


Figure 17. Contact forces responsible for convergence along the degree of freedom x

Orientation correction using impedance control

In order to correct the rotation on a given plane, it is possible to force a collision on a certain direction, while the control loop of the rotation axis is more relaxed than the remaining degrees of freedom. With this configuration, the manipulator will rotate around the relaxed degree of freedom until it is aligned with the collision surface (Figure 18).

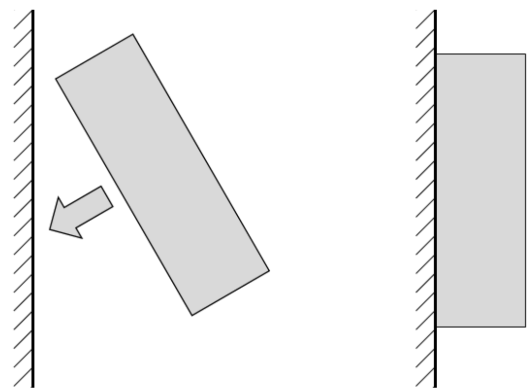


Figure 18. Orientation correction using impedance control

The implementation of this strategy for the studied assembly task can be accomplished using the cartesian impedance controller, configured to the minimum stiffness value for rotation around the z axis. The collision can be forced between the bearing cap and one of the engine block's exterior walls. A shear force condition should be programmed to stop the collision motion when a certain force threshold is measured, thus ensuring the successful orientation correction.

Environment exploration through collision detection

This strategy consists of exploring the position of a fixed component on the assembly station by probing the environment until a predetermined collision with the workpiece is detected. In cases with no orientation error, the existing errors could be completely eliminated by using this strategy one time for each degree of freedom subjected to errors.

However, rotational errors have a direct influent on this strategy's capability to correct translation errors. This happens because the reference frames for the probing motions (reference frames located at tool level) are not correctly aligned with the workstation and, thus, the motions will move away from the perfect projected trajectory. Figure 19 displays how rotational errors decrease the effectiveness of this strategy.

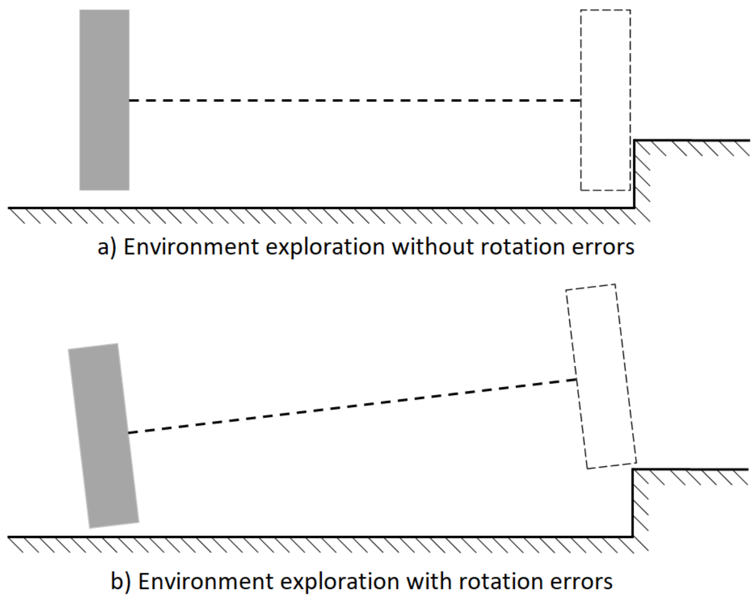


Figure 19. Negative impact of orientation errors on environment exploration

Another factor that impacts the effectiveness of this strategy is the distance between the start position of a probing motion and the expected contact position (size of the probing motion). This factor does not have a direct influence on the strategy, but it amplifies the errors caused by orientation uncertainties. Figure 20 shows how the length of the probing motion amplifies the impact of orientation errors on this strategy.

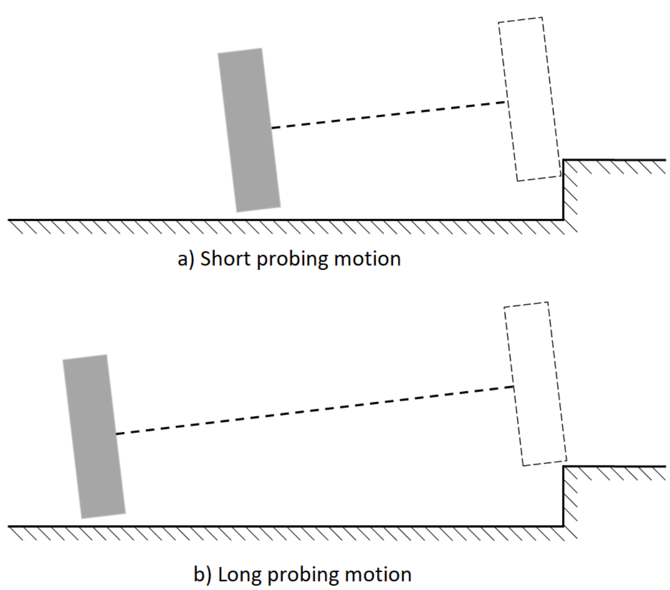


Figure 20. Amplification of the impact of orientation errors on environment exploration

As a result, this strategy can only be exclusively used to fully correct translation errors for situations with no orientation uncertainties. In the scope of the present study, since the assembly routines must also surpass orientation errors, this strategy can only be used to decrease translation errors, but never to fully eliminate them. Furthermore, it is possible to complement this strategy with a previous orientation correction, thus eliminating their influence on the obtained results.

Blind search using oscillations

This strategy was developed as a way of exploring the "Cartesian impedance controller with overlaid force oscillation" of the LBR iiwa, exposed in the Subsection 3.4.1., which implements an overlaid force oscillation to one degree of freedom while controlling the manipulator in cartesian impedance control. Consequently, this method indirectly implements positional oscillations, whose amplitudes are reliant of the selected force amplitude and the configured stiffness for that degree of freedom.

The implementation of this strategy allows the execution of blind search routines with restricted contact forces. In fact, since the oscillations are created through a force overlay, the contact force in that degree of freedom will never surpass the overlaid force amplitude.

Moreover, it is important to note that there might be a discrepancy between the expected theoretical position oscillation and the actually measured oscillation. This inconsistency arises from the inaccuracies of the force measurement method implemented in the manipulator (estimated using joint torque as opposed to measured). It is also expected that this positional inaccuracy should decrease with the implementation of bigger force overlays, since the relative impact of the measurement errors decreases with the increase of the force magnitudes.

In order to test this hypothesis, a few tests were performed, with increasing values of force overlay and a fixed cartesian stiffness value. The positional oscillation amplitude was measured using the inherent data recording capabilities of the manipulator.

Table 5. Impact of the overlaid force amplitude on the relative error of the oscillation amplitude

Force Overlay $[\boldsymbol{N}]$	**Stiffness** $\left[\frac{\boldsymbol{N}}{\boldsymbol{m}}\right]$	**Force Bias** $[\boldsymbol{N}]$	**Theoretical oscillation** $[\boldsymbol{mm}]$	**Measured oscillation** $[\boldsymbol{mm}]$	**Relative error**
10	1000	0.5	10	4.4	56 %
25	1000	−0.5	25	16.6	34 %
50	1000	−0.8	50	40.1	20 %
100	1000	−2.5	100	86.9	16 %

As expected, the test results listed in Table 5 indicate that the error of the measured oscillation amplitude relative to its theoretical value decreases with the increase of force overlay. Furthermore, the measured results highlighted the need to overlay an additional constant force to ensure that the oscillation was symmetrical in both directions – Force Bias represented in Table 5.

Additionally, the significant error values displayed in the above table emphasize the need to measure and control the oscillation amplitude, as opposed to directly applying the expected theoretical values.

4.4.2. Developed search methods

Oscillation method

This method was developed to overcome small translation and orientation uncertainties. It implements the strategies of blind search using oscillations and direct impedance control convergence. This approach executes a continuous vertical motion while employing the different oscillations and control modes required to complete the assembly task.

Firstly, it performs an oscillation on the y axis to correct the error of this degree of freedom. This correction is materialized by the slotting of one edge of the bearing cap inside the rod bearing journal of the crankshaft.

Subsequently, an oscillation is made around the z axis, while simultaneously keeping a low stiffness value for the x axis. This translates in an orientation correction through oscillation (to ensure that the bearing cap fully enters the required slot) and a simultaneous natural impedance convergence along the x axis.

Thus, this method corrects errors on all three required degrees of freedom. The correction of the translation along the y axis and orientation around the z axis is accomplished through the oscillation strategy, whereas the translation along the x axis is achieved through direct impedance control convergence.

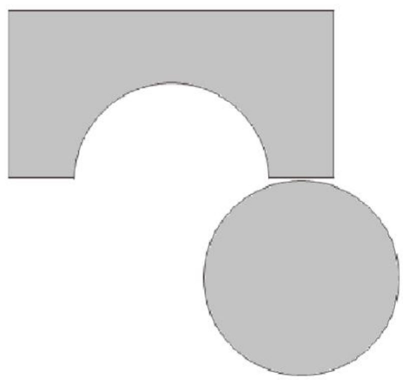
The configuration of the oscillations must follow some basic rules to ensure the successful accomplishment of the task:

- The oscillation amplitude must be bigger than the maximum possible error for that degree of freedom (larger than the positional uncertainties on that degree of freedom).
- The frequency of the oscillation must be synchronized with the velocity of the downward motion to ensure that the full range of translation values are swept when the workpiece is close to entering the slot.

Additionally, in order to guarantee the success of this method, it is important to keep in mind the geometrical characteristics of the different components and how they may limit its execution. In fact, it is evident that, for the studied assembly task, the geometrical traits of the components limit the maximum translational errors that this approach can overcome:

a) The thickness of both the rod bearing journal walls and the bearing cap play a major role in the maximum oscillation amplitude along the y axis, as it is necessary to ensure that the oscillations do not cause the manipulator to exit the assembly region. Figure 21 a) shows an implementation of this method that did not account for this characteristic, resulting in an unsuccessful assembly task.
b) Since the error along the x axis is surpassed by direct impedance control convergence, it is necessary to ensure that the contact point is done in the correct side of the cylinder curvature (as displayed in Figure 17), thus limiting the maximum admissible range for errors along the x axis and avoiding situations such as the one displayed in Figure 21 b).

a)



b)

Figure 21. Failed oscillation methods due to unaccountability of geometrical traits

Another important factor to keep in mind for the appropriate configuration of the oscillations is the admissible collision force. Most assembly tasks benefit from low contact forces, especially when working with small and fragile components. As such, since the components of the workpiece are not particularly delicate, the maximum values for contact force and toque were defined at 12.5 N and 2 Nm, respectively. Numerous different values of stiffness were tested for the mentioned force and torque overlays and the respective oscillation amplitudes were measured using the data recorder of the manipulator. The results are presented in Tables 6 and 7.

Table 6. Oscillation amplitude along the y axis for different values of stiffness

Force Overlay $[N]$	**Stiffness** $\left[\frac{N}{m}\right]$	**Force Bias** $[N]$	**Theoretical oscillation** $[mm]$	**Measured oscillation** $[mm]$
12.5	1000	1.0	12.5	6.0
12.5	900	0.7	13.9	6.5
12.5	800	0.8	15.6	7.2
12.5	700	0.4	17.8	8.1
12.5	600	0.2	20.8	9.2
12.5	500	0.1	25.0	11.0
12.5	400	0.0	31.2	13.3

Table 7. Oscillation amplitude around the z axis for different values of stiffness

Torque Overlay $[N \cdot m]$	**Stiffness** $\left[\frac{N \cdot m}{rad}\right]$	**Torque Bias** $[N \cdot m]$	**Theoretical oscillation** [°]	**Measured oscillation** [°]
2	40	−0.2	2.9	2.7
2	30	−0.2	3.8	3.5
2	20	−0.1	5.7	5.0
2	15	−0.1	7.6	6.5
2	10	0.0	11.5	9.2

This data was crucial for testing the limits of this method as it allowed small adjustments to the oscillations while testing different error values. These tests also allowed the establishment of admissible translational error ranges for this method, whose values are summarized in Table 8.

Table 8. Admissible translation error ranges for different values of orientation errors for the oscillation method

Tested orientation errors	**Admissible error ranges**	
	$\boldsymbol{x}$ **axis**	$\boldsymbol{y}$ **axis**
$\alpha_A = 0°$	$x_A \in [-13, 13]\ mm$	$y_A \in [-12, 12]\ mm$
$\alpha_A = 2° \vee \alpha_A = -2°$	$x_A \in [-10, 10]\ mm$	$y_A \in [-10, 10]\ mm$
$\alpha_A = 4° \vee \alpha_A = -4°$	$x_A \in [-10, 10]\ mm$	$y_A \in [-10, 10]\ mm$
$\alpha_A = 6° \vee \alpha_A = -6°$	$x_A \in [-9, 9]\ mm$	$y_A \in [-9, 9]\ mm$

Once again, it is relevant to underline that these values of maximum admissible errors are reliant on the specific geometry of the studied assembly task and not on the direct capabilities of the employed strategies. Thus, this method could be implemented for larger admissible error ranges if the geometrical traits of the components allowed it.

Furthermore, values of orientation error larger than 6° tremendously impacted the effectiveness of this method, not only because they would require large orientation oscillations, but also because it affects the direction of the oscillations along the y axis (since all motions are planned relative to reference system hierarchically bellow the manipulators flange).

Figure 22 displays a graph representing the evolution of the manipulator's position while performing this method. It allows an effortless visualization of the different segments of the described method.

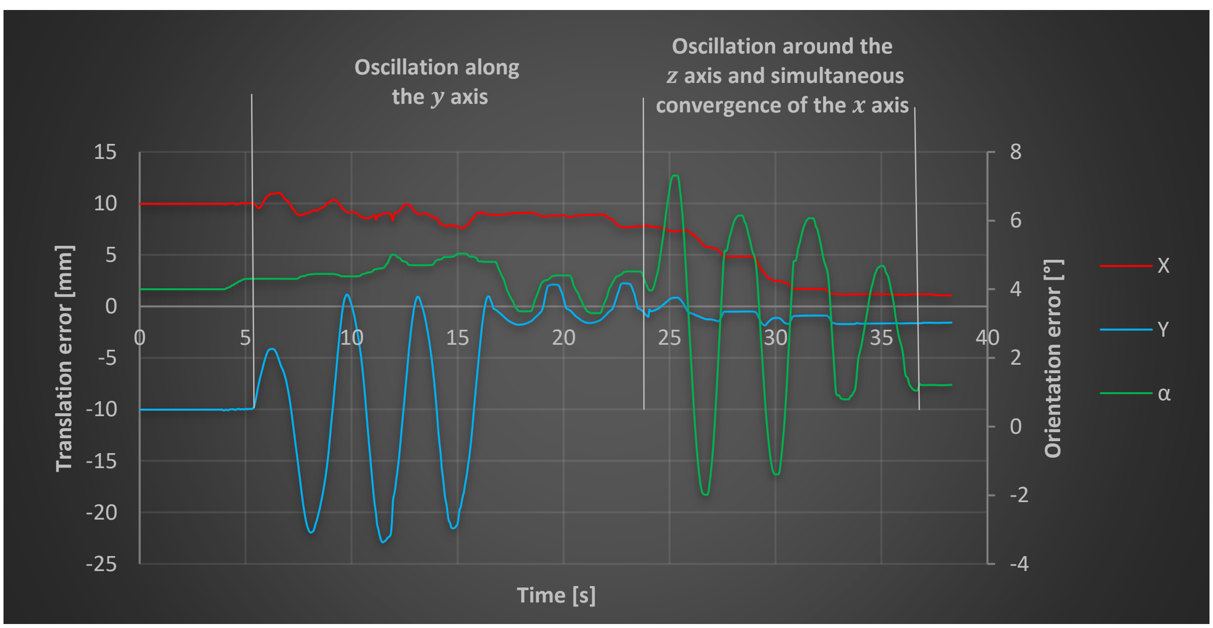


Figure 22. Graph showcasing the evolution of the manipulator's error during the oscillation method

Oscillation method complemented by environment exploration

This approach aims to increase the admissible translation errors by complementing the oscillation method with a simple environment exploration strategy, in which two probing motions are performed on the side of the engine block to locate its position (in the x and y axis). However, since the orientation uncertainties impact the trajectory of the motions, the perceived location of the engine block may not accurately reflect its actual position. As such, it is expected that the strategy of environment exploration will not be sufficient to fully eliminate the present translation errors and complete the assembly tasks. Instead, it can be used to reduce said errors to values within the admissible range of the oscillation method.

Figure 23 displays the selected contact positions implemented in this method.

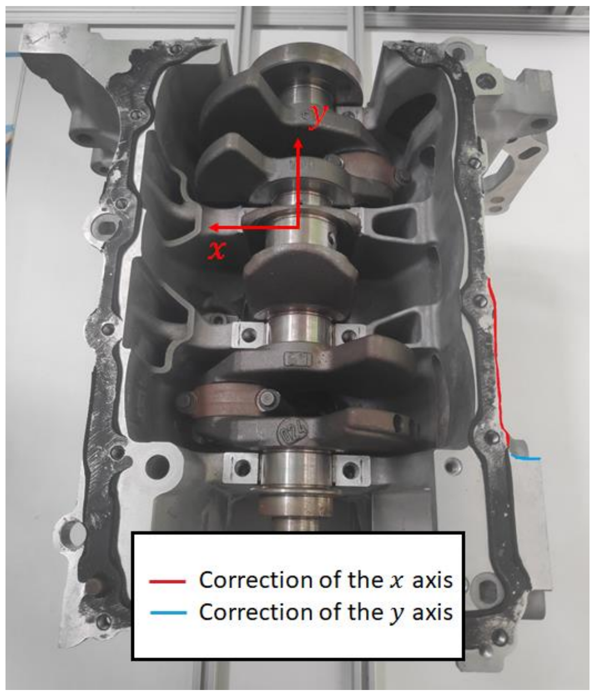


Figure 23. Selected contact positions for the environment exploration strategy

This simple exploration method showed great potential for decreasing the positional errors of the task. For maximum orientation errors of $\pm 4°$, testing revealing that this method was capable of decreasing the maximum registered errors in the following manner:

$$\begin{cases} x_A \in [-30, 30]\ mm \\ y_A \in [-30, 30]\ mm \\ \alpha_A \in [-4, 4]\ ° \end{cases} \rightarrow \begin{cases} x_A \in [-7.9, 7.9]\ mm \\ y_A \in [-8.9, 8.9]\ mm \\ \alpha_A \in [-4, 4]\ ° \end{cases}$$

The maximum error range after the correction is set within the maximum admissible error values for the oscillation method, thus proving to be a good strategy for situations where the translation uncertainties of the assembly task are greater than those tolerated by the former method.

Additionally, the distance from the assembly position to the contact probe position amplifies the trajectory errors caused by orientation errors, as exposed in Figure 20. Therefore, it is expected that the initial correction by environment exploration is more effective in situations where the geometric characteristics of the workspace allow the exploration to be done closer to the target assembly position.

Oscillation method complemented by orientation correction

This approach was developed in order to increase the admissible orientation errors of the oscillation method by performing a previous orientation correction on the side of the engine block. After performing the orientation correction, the manipulator returns to the intended assembly position, preserving the corrected orientation value, where it is expected to be capable of successfully implementing the oscillation method.

Figure 24 shows an above view of the lateral wall of the engine block, where the orientation correction using impedance control takes place.

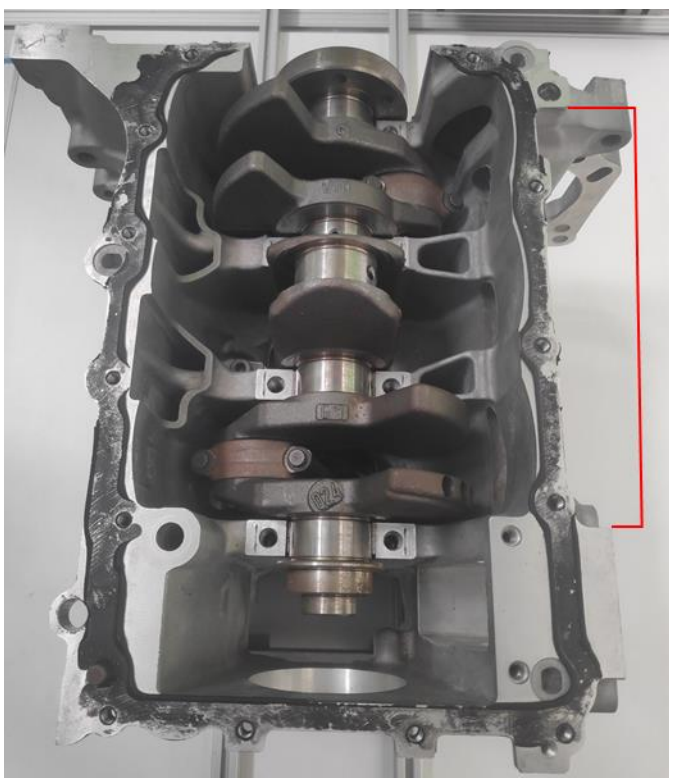


Figure 24. Segment of the engine block where the orientation correction takes place

The considered wall is not perfect for the given task as has numerous small protrusions, making it impossible to fully rectify the orientation with just one correction motion. However, it is the best possible position of the available options, since the other walls either have bigger protrusions or inadequate geometries.

This strategy was tested for increasing orientation errors and the range of admissible translation errors for the oscillation method. It was concluded that the performed orientation correction is capable of narrowing the maximum uncertainties values in the following manner:

$$\begin{cases} x_A \in [-10, 10]\, mm \\ y_A \in [-10, 10]\, mm \\ \alpha_A \in [-20, 20]\,^\circ \end{cases} \rightarrow \begin{cases} x_A \in [-10, 10]\, mm \\ y_A \in [-10, 10]\, mm \\ \alpha_A \in [-3, 3]\,^\circ \end{cases}$$

Since the range of maximum errors after correcting the orientation falls within the admissible error range for the oscillation method, it is possible to infer that the implementation of orientation correction by impedance control was successful and is a viable strategy to increase the admissible orientation errors of a given method.

Environment exploration complemented by orientation correction

In the previously developed methods, it was proven that environment exploration and orientation correction strategies could be used to reduce position and orientation errors, respectively. However, these strategies were limited since orientation errors had a negative impact on the environment exploration and positional uncertainties did not allow a consistent orientation correction on the uneven wall used for the rectification.

Since the shortcomings of one strategy can be minimized by the employment of the other, it is feasible to construct the theory that both strategies could be implemented together to fully eliminate the uncertainties on all degrees of freedom. Thus, a method using both strategies was developed to test this hypothesis, culminating on a routine without the need of oscillations to complete the assembly task.

This approach is reminiscent of an iterative process because each time the orientation correction strategy is executed, the orientation uncertainties decrease, which in turn allow environment exploration routines to achieve narrower ranges of positional uncertainties. Hence, it is possible to alternate between these two methods, until the exact position and orientation of the engine block are known, allowing the effective completion of the assembly task.

Furthermore, the developed strategy for orientation correction inadvertently performs environment exploration on one degree of freedom as the rectification is performed through contact on the side of the engine block. For this reason, it is more efficient to start the routine by performing a rectification of the manipulator's orientation and the explicit environment exploration routines only needs to be performed for the degree of freedom y.

Figure 25 displays the admissible area for the first point of contact – as long as the first contact is done in this region, the algorithm ought to be able to correct the present errors. The exact position of the first contact is reliant on both the translational and the orientation errors.

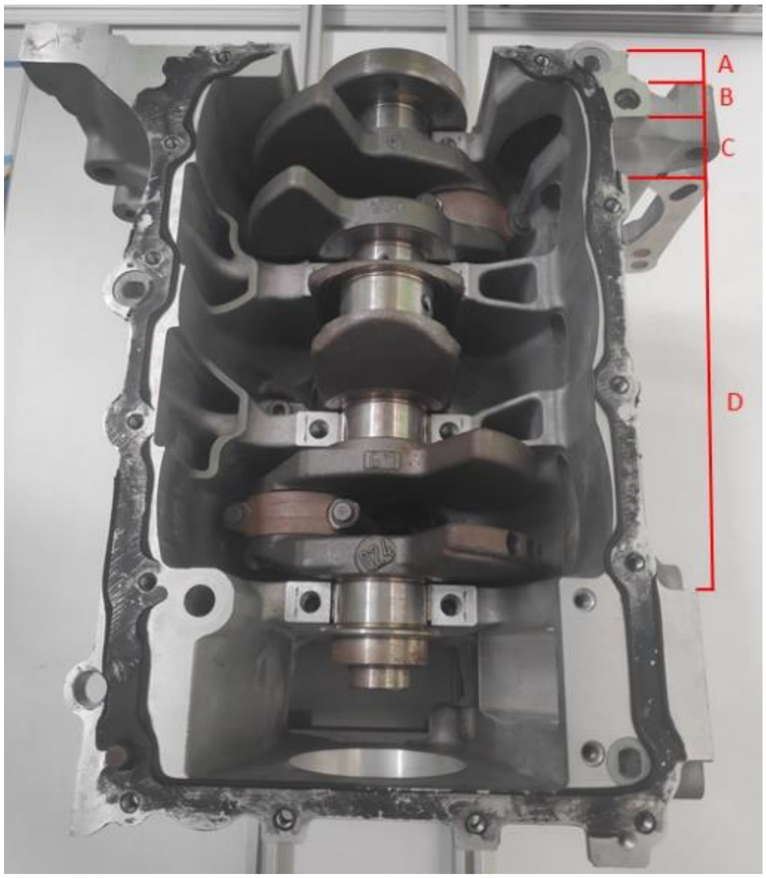


Figure 25. Different admissible locations for the landing of the first probing motion

Four different contact regions were identified – depending on the landing position, different motions for the exploration of the environment must be applied. Since positions C and D cover most of the possible landing region, they correspond to the contact position of most error values. Therefore, these positions were approached as a base case, i.e., if it is detected that the first probing motion landed in position A or B, a correction should be performed to apply the base case. However, to increase efficiency, the performed correction should be different for the position A and B.

In order to quickly identify cases landing on position A and B, the first orientation correction was analyzed for these cases. This evaluation revealed that, when the bearing cap lands in these positions, the registered orientation correction was always larger than 17°. However, even though position A and B imply a correction of more than 17°, the opposite may not be true. In fact, for situations with a high orientation and translation errors, the same condition may be met, even if the probing motion lands in Position D – these situations correspond to false positives situations.

Since the rotation condition is not enough to identify positions A and B, further position tests were developed. These examinations aim to check false positive conditions; thus, they should only be carried out when a high rotation correction is identified. These tests are a variant of the environment exploration strategy since they use collision detection to verify the existence of geometrical landmarks of the engine block in order to discover its relative position. Figure 26 displays the geometrical characteristics of the engine block used for the configuration of these conditions.

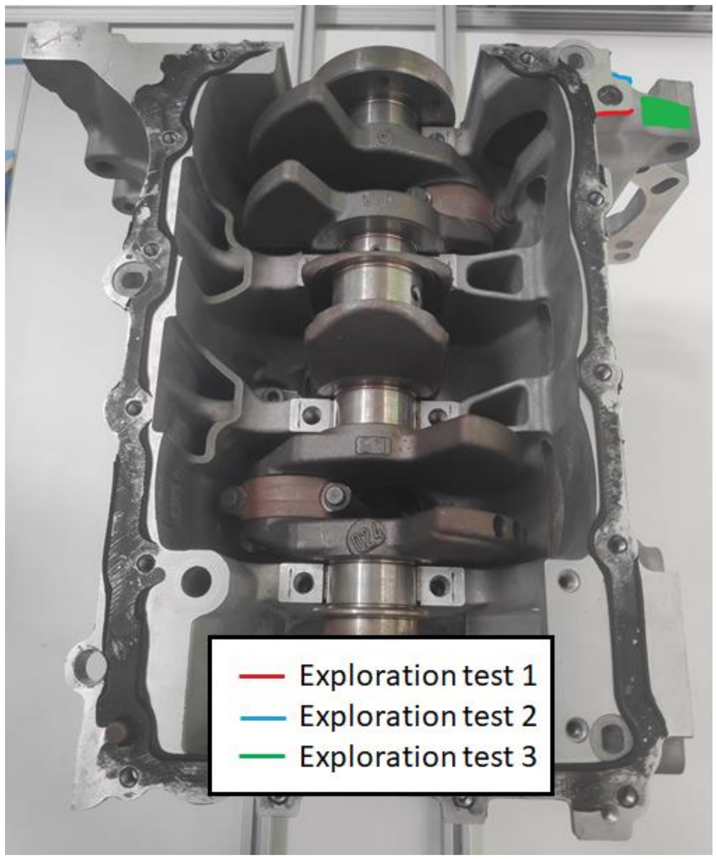


Figure 26. Defined environment exploration tests

With the described tests, it is possible to create a simple algorithm capable of identifying the landing position of the probing motion and calling the corresponding correction and base case routines. Figure 27 demonstrates a flowchart of this algorithm.

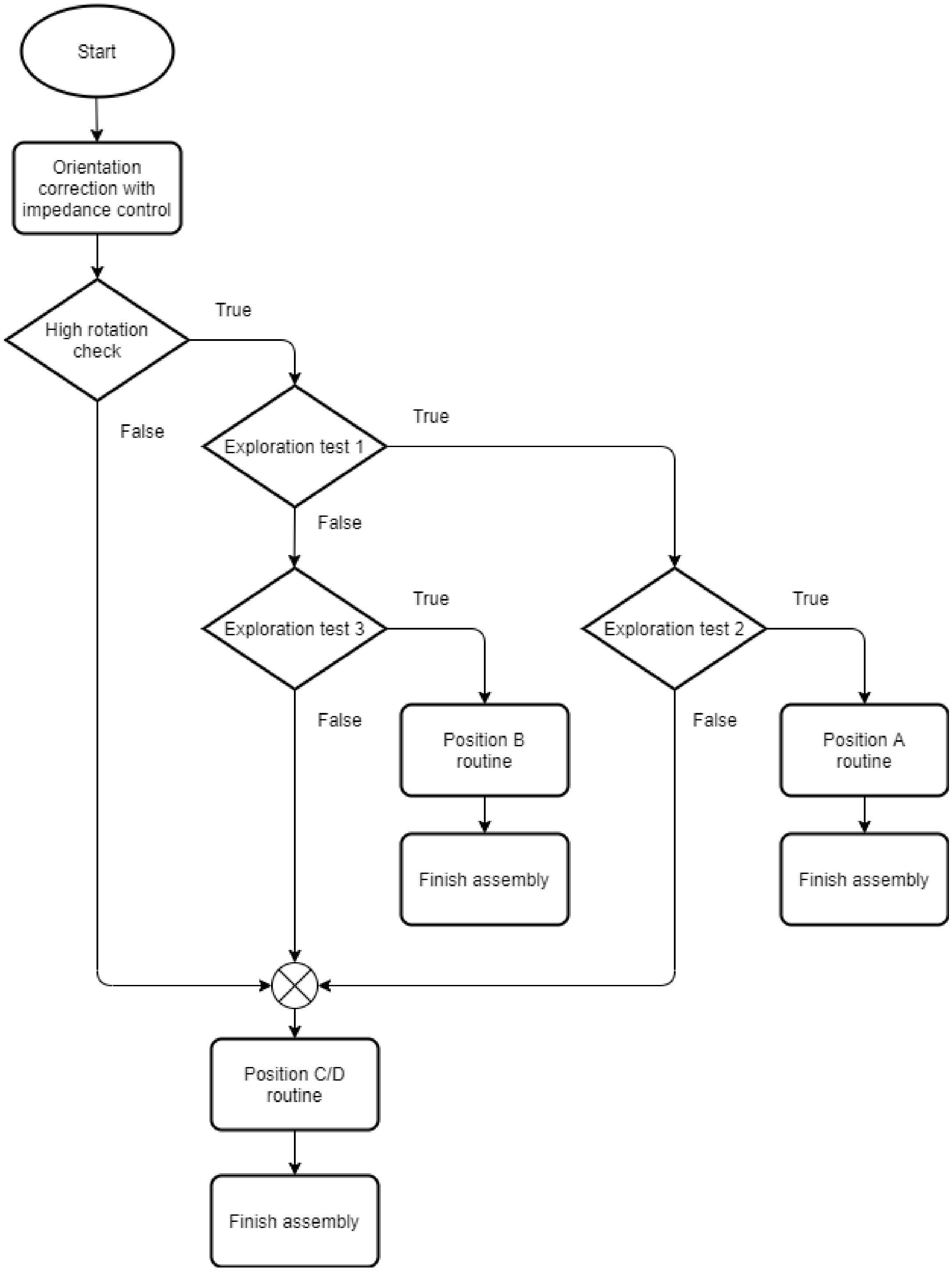


Figure 27. Flowchart showcasing the "Environment exploration complemented by orientation correction" method

The action blocks "Position A routine" and "Position B routine" correspond to the respective correction to the position C/D area and calling of the base case procedure, whereas the action block "Position C/D routine" only calls the base case.

The base case itself works as a series of orientation corrections and environment explorations on the y degree of freedom. In order to correctly and reliably rectify the orientation, the same location of the engine block wall must be used. However, two distinct zones were defined for this correction in order to increase its overall efficiency. The algorithm selects the zone that is most likely closer to, based on the value of orientation correction performed (assuming that translational values are randomly distributed around the perfect assembly position, i.e. are not biased). Figure 28 shows the defined contact areas for orientation correction and environment exploration on the y degree of freedom of the base case.

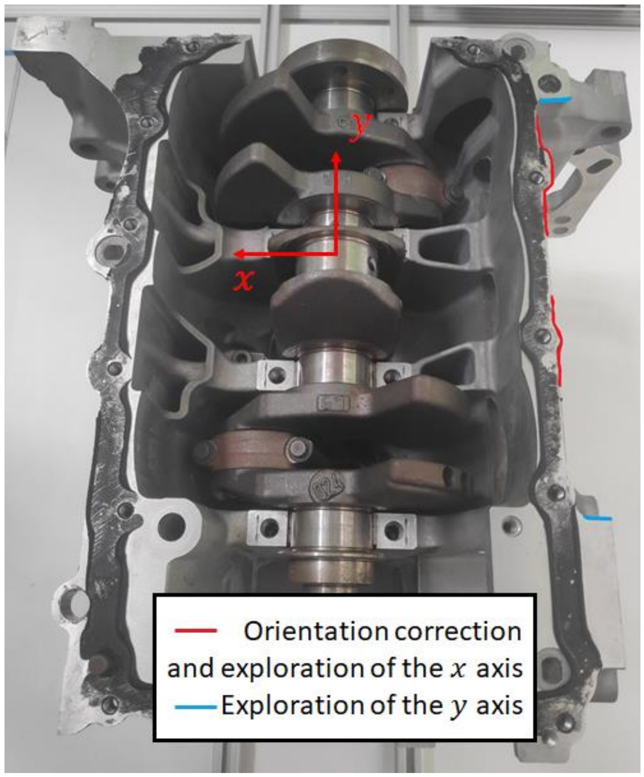


Figure 28. Locations for the orientation correction and environment exploration of the base case

After fully eliminating the errors on all degrees of freedom, the manipulator can identify and approach the perfect assembly position. Subsequently, the "Finish assembly" block can be called, in order to execute a direct downward motion until it senses a vertical collision, indicating that the bearing cap reached its final position and the assembly was completed.

This method was able to fully eliminate positional errors within the following ranges:

$$\begin{cases} x_A \in [-30, 30]\ mm \\ y_A \in [-30, 30]\ mm \\ \alpha_A \in [-20, 20]\ ° \end{cases}$$

However, since the only required condition for the success of this method is for the first probing motion to land inside position A, B, C or D, the admissible error ranges can be arranged depending on the applications needs. As Figure 29 suggests, if the maximum orientation error decreases (represented by the red lines), the translation errors that this method can overcome increase (as displayed by the blue squares).

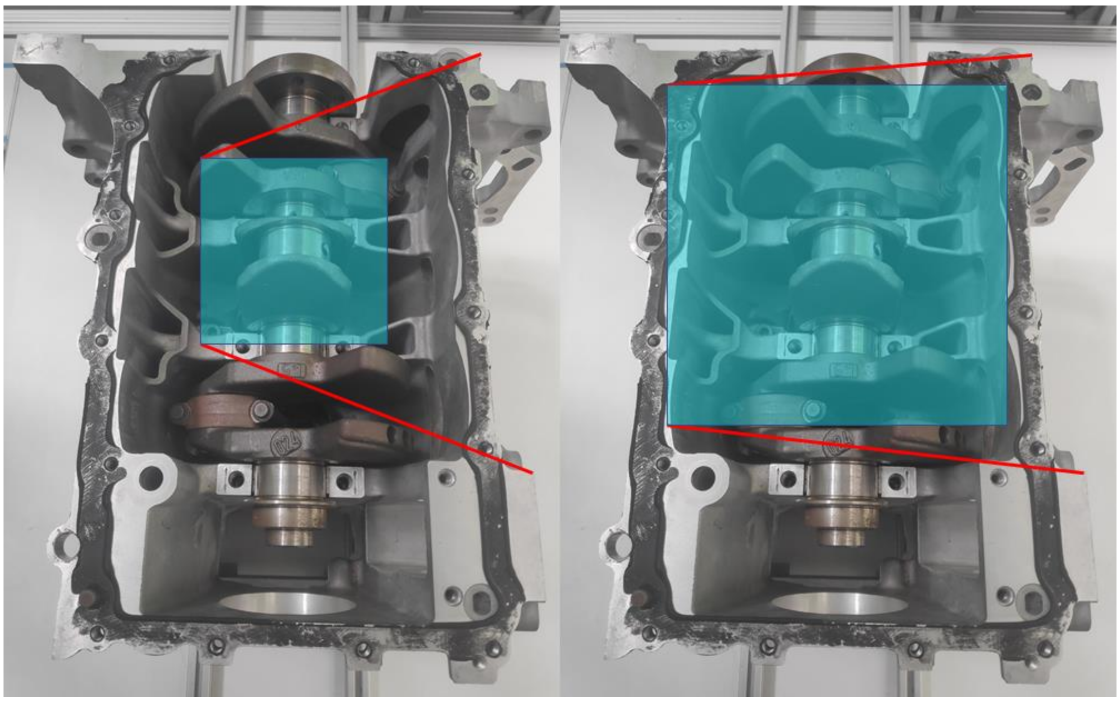


Figure 29. Interchangeability of translation and orientation errors

Moreover, if more than one task is to be performed on the engine, this routine only needs to be performed once to fully determine the engine position. Once all errors are eliminated, the manipulator is able to accurately position itself to perform assembly tasks in every position of the engine, which reinforces the importance and flexibility of this method.

Developed methods summary

Table 9 summarizes the surpassable error ranges by each developed search method. Once again, these values were measured in the assembly workspace, relative to the perfect assembly position and do not reflect the errors on the mobile robot workspace.

Table 9. Developed search methods surpassable error ranges

	x_A	y_A	α_A
Oscillation method	$[-10, 10]\ mm$	$[-10, 10]\ mm$	$[-4, 4]\ °$
Oscillation method complemented by environment exploration	$[-30, 30]\ mm$	$[-30, 30]\ mm$	$[-4, 4]\ °$
Oscillation method complemented by orientation correction	$[-10, 10]\ mm$	$[-10, 10]\ mm$	$[-20, 20]\ °$
Environment exploration complemented by orientation correction	$[-30, 30]\ mm$	$[-30, 30]\ mm$	$[-20, 20]\ °$

4.5. Solution generalization

The proposed search methods were able to overcome positional errors and perform an assembly task with the corresponding workpiece. However, since these methods are too reliant on the workpiece's geometries, they cannot be directly integrated with other assembly stations. For this reason, this section aims to propose tools for the implementation of the developed strategies in generic assembly stations.

The pure oscillation method is generic enough that it can be applied directly to numerous assembly stations with only a few calibrations (oscillated degrees of freedom and oscillation amplitude), whereas the environment exploration method is much more task focused since it is highly reliant on the geometry of the assembly parts.

Furthermore, oscillation-based methods accomplish the assembly through blind positioning without complete error correction and, therefore, the routine must be performed for every individual assembly task. On the other hand, the "Environment exploration complemented by orientation correction" method fully eliminates the errors in the assembly workspace, which means that it can perform multiple assembly tasks on the same station with only one correction routine. Therefore, oscillation-based methods are more suited for individual assemblies, whereas the environment exploration method is essential to perform multiple assembly tasks at the same station.

In order to develop a solution as generic as possible, it is important to account for situations where multiple assemblies are performed at a single station. Thus, these proposed solutions will focus on the "Environment exploration complemented by orientation correction" method. Nevertheless, a few considerations will also be done for the oscillation strategy.

4.5.1. External accessory for planar error correction

One solution for the implementation of the “Environment exploration complemented by orientation correction” method on generic assembly tasks is the development of an external accessory that can be mounted on a fixed location relative to the assembly station (e.g. on the side of the engine block). This accessory serves as the contact point for all the environment exploration and orientation correction motions, thus eliminating the need of identifying potential contact positions within the station. Moreover, this solution allows the implementation of this method under perfect conditions, allowing the execution of a faster, optimized routine for general assembly tasks.

The experience acquired during the development of the aforementioned search methods was essential for identifying the required characteristics of the external accessory. As such, the optimal design solution for this accessory is two orthogonal surfaces (Figure 30) – one for orientation correction and exploration of one degree of freedom and the other for exploration of the other degree of freedom.

Due to the nature of the probing motions, these surfaces have different length requirements. The former has a variable length requirement since it must ensure that the first probing motion lands on it, which is impacted by the required range of surpassable errors. On the other hand, since the first contact fully corrects the orientation and one degree of freedom, the length of the other surface can be constant (not larger than $\approx 20\ mm$).

Additionally, it is important to state that an oversized accessory can be used in situations with low error without penalizing the cycle time of the routine, whereas an undersized accessory cannot be used to guarantee the success of the routine. For that reason, if an accessory is designed for use in multiple assembly stations, it is preferable that it is oversized.

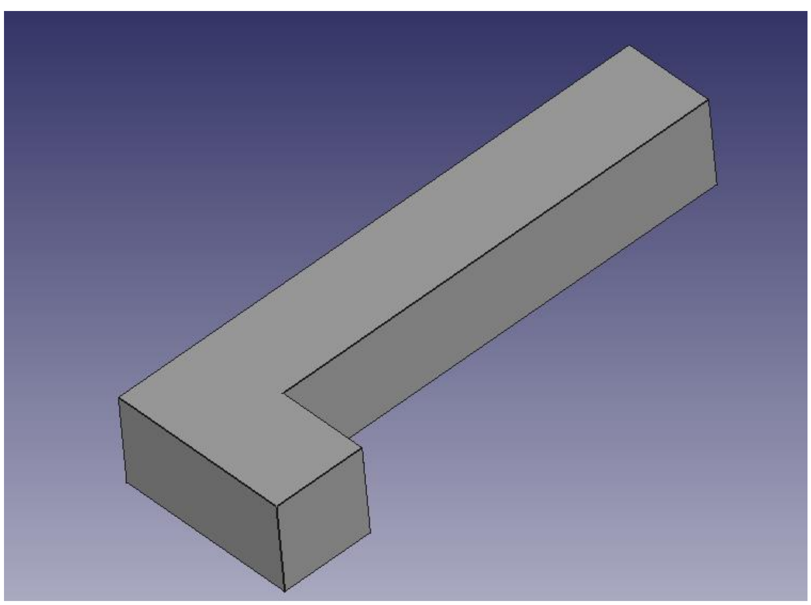


Figure 30. Optimal design for the external accessory

The fixation system of this accessory should be designed to accommodate the specific assembly station that it is mounted on. It can be mounted directly on the assembly station floor or on a lateral wall of a given workpiece. The only restriction is that it is mounted on a fixed location, parallel to the ${}_xO_y$ plane.

Furthermore, since the optimal design of this accessory is so simple, it is possible to improvise two orthogonal surfaces with available materials in the station. Figure 31 shows an improvised implementation of this accessory using two beams in a fixed location relative to the engine block.

Figure 31. Example of an improvised external accessory

4.5.2. CAD analysis algorithms

An alternative solution for the generalization of the proposed methods is the development of an algorithm capable of analyzing the assembly station CAD file. The goal of this algorithm is to identify reachable orthogonal surfaces, where the “Environment exploration complemented by orientation correction” method can be implemented.

At least one of the identified surfaces should be as long as possible, as this length is crucial for the implementation of large ranges of surpassable errors. Furthermore, it is preferable that the larger surface is flat (i.e. does not have any protrusions) in order to fully correct the orientation with just one motion (instead of the iterative approach performed in the subsection 4.4.2.). Additionally, it is essential that the identified surfaces are reachable by the manipulator, i.e. the probing motions of these surfaces cannot be obstructed by other components.

This method would allow the implementation of the “Environment exploration complemented by orientation correction” method without the need of an external accessory. However, since this tool only identifies the already available surfaces, it is limited to the characteristics of each assembly station – there might be no available orthogonal surfaces that fulfill all the listed requirements. Thus, this method is extremely limited in relation to the previous solution, as it offers no guarantees that the necessary errors would be surpassed.

Moreover, CAD analysis could also be integrated to automatically determine the geometric limitations of the oscillation method – the maximum oscillation value that could be integrated without compromising the success of the task.

To sum up, the described CAD analysis algorithms have the advantage of not needing any extra components than the required assembly parts but its implementation does not ensure the success of any general assembly. Instead it can be seen as a way of automating the process performed in the subsection 4.4.2., where the geometrical characteristics of the assembly station were manually examined to find an optimal strategy for compliant error correction.

4.5.3. Gripped workpiece

Another crucial factor for the generalization of the proposed search methods is the geometry of the gripped workpiece. This issue does not have a major direct impact on the environment exploration but is essential for the orientation correction motion. Figure 32 displays the orientation correction process for different profiles of the gripped workpiece.

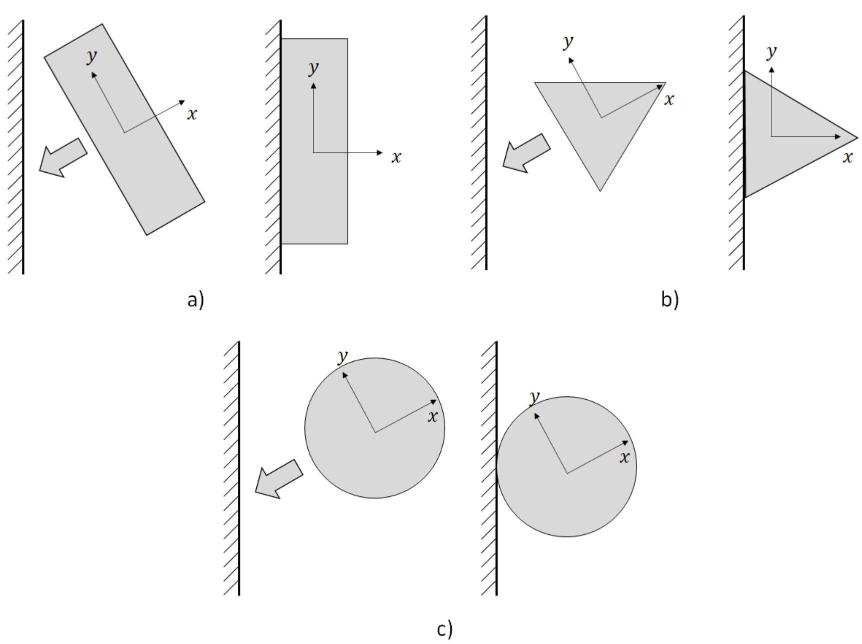


Figure 32. Influence of gripped workpiece geometry on orientation correction

It is clear that the developed method is only capable of correcting the orientation when the gripped workpiece has, at least, one flat surface. Furthermore, larger surfaces lead to orientation corrections with lower contact forces, which might be beneficial for the routine. This is a limitation of the developed method since it cannot perform the correction for every single workpiece (e.g. it cannot correct the orientation with cylindrical parts).

However, since the described routine fully eliminates the present errors, it is quite simple to overcome this limitation by previously running the routine with an adequate workpiece. As an example, if two assemblies are to be performed on a given station, one with a cube and another with a cylinder, the routine should be performed with the cube and then the cylinder can be directly assembled. Moreover, if the assembly order requires the cylinder to be mounted first, the routine can still be performed with the cube in order to fully locate the assembly station and, subsequently, the assembly can proceed in the correct order.

This strategy can be scaled to generic applications with the use of a dedicated workpiece for the completion of the correction routine (e.g. a rectangular prism). In fact, the use of a dedicated workpiece increases the flexibility of the system since the success of this strategy will not be compromised, even if none of the parts to be assembled at a given station have an appropriate geometry for the completion of the orientation correction.

Alternatively, if it is required to perform the correction directly with a workpiece without a flat surface, another mechanism of orientation correction must be developed (instead of the used orientation correction by impedance control) to complement the environment exploration method.

5. Solution implementation

This chapter describes the implementation of the proposed solution on the constructed mobile assembly station. Initially, the main characteristics of the setup calibration will be outlined and the error correlation between the assembly workspace used to develop the methods and the mobile robot workspace will be established. Then, the selected implemented methods will be mentioned and quantified in both workspaces. Furthermore, a description of how the flexibility of the station can be increased by combining multiple search methods will be delineated. Finally, the architecture of the system will be briefly explained as a whole, emphasizing the communication between the mobile robot and the manipulator.

5.1. Mobile robot setup

After the testing of all developed methods, the manipulator was mounted on top of the mobile robot built in the scope of the ScalABLE4.0 project (Figure 33). In order to implement the developed methods, a stopping position for the mobile robot must be selected and the appropriate motion frames of the created algorithms must be calibrated (this step is crucial since the manipulator is now on a different position relative to the engine block).

Once the robot leaves the calibration area, it is not possible to move it back to this position as it will always be influenced by the location errors of the positioning system. The only way of guaranteeing that the robot is in the perfected calibrated position is to calibrate it again in the wanted configuration.

Figure 33. Mobile robotic assembly station

In the development setup, all the position and orientation errors were imposed, since the perfect assembly position was always known. However, in this setup, all the position and orientation errors arise from the uncertainties of the navigation system of the mobile robot and, thus, cannot be easily measured.

This difference of approach implies that the developed methods were created and tested using error values relative to a reference frame located in the assembly workspace, whereas the implemented system's errors arise from the positioning system of the mobile robot and, thus, the errors of this setup should be quantified in a reference frame relative to the mobile robot workspace. Therefore, to quantify the errors of the mobile robot workspace that the developed methods can overcome, it is crucial to establish a correlation between these workspaces.

Figure 34 shows a representation of both workplaces. The reference frames A and R correspond to the calibrated assembly and robot workspaces, respectively. In turn, the reference frame R' corresponds to a generic position of the robot, with a certain error relative to R. Finally, the frame A' corresponds to the calibrated assembly position when the robot is in the generic position R'.

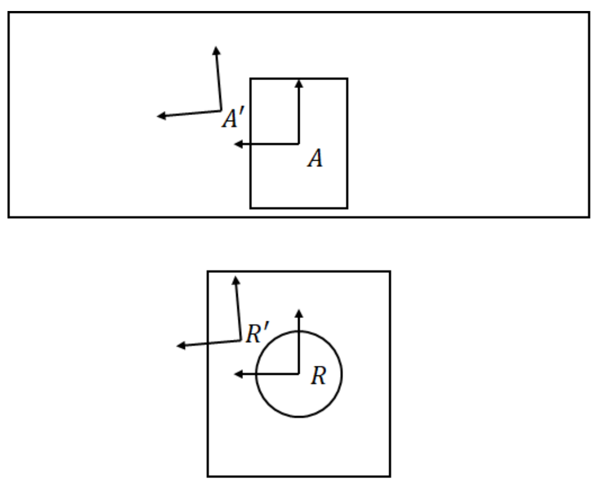


Figure 34. Assembly and mobile robot workspaces

The error of the generic robot position relative to the calibrated position can be measured in the reference frame R by x_R, y_R and α_R. As displayed in Figure 35, it is possible to express the relationship between the frames A and A' using the values of the error between R and R' (express the error in the assembly workspace, using the error of the robot workspace as variables).

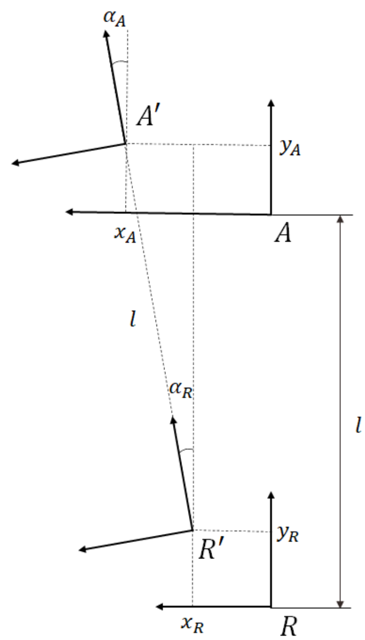


Figure 35. Correlation between assembly and mobile robot workspaces

$$\begin{cases} x_A = x_R - l \cdot \sin \alpha_A \\ y_A = y_R - l \cdot (1 - \cos \alpha_A) \\ \alpha_A = \alpha_R \end{cases} \tag{3}$$

From this set of equations, it is possible to infer that:

- A translation error on the robot workspace results in the same translation on the assembly workspace.
- An orientation error on the robot workspace results in the same orientation error on the assembly workspace.
- An orientation error of the robot also causes a translation error on the assembly workspace:
    - The corresponding translation might be positive or negative relative to the degree of freedom x.
    - The corresponding translation is always negative relative to the degree of freedom y.
    - For small orientation errors, the corresponding translation error relative to y is almost negligible when compared to the translation error relative to x.
- The distance, l, between the manipulator and the assembly position amplifies the translation errors caused by orientation errors.

The chosen stopping position of the robot during its calibration process has a direct impact on the amplification of translation errors on the assembly workspace caused by orientation errors of the mobile robot (as it directly defines the distance l). As an example, if the distance l is set to $1000\,mm$, an orientation error of $1°$ on the robot workspace causes a translation error of $x_A = 17.45\,mm$ and $y_A = 0.15\,mm$ on the assembly workspace.

These considerations are crucial as they were the basis for the collaboration with the mobile robot team. It is essential that the orientation error is as low as possible, even if that implies slightly larger translation errors. Furthermore, in order to minimize the influence of the orientation errors of the robot on the assembly workspace, the selected calibration position should be as close to the assembly station as possible. Nevertheless, this distance should still be large enough to ensure that the navigation system of the robot has enough accuracy to avoid collisions with the assembly station. For these reasons, the defined value of this distance for the calibration position was $l = 500\,mm$.

5.2. Implemented methods

Preliminary tests of the positioning system of the mobile robot revealed that it could routinely complete the solicited trajectory, positioning the robot with orientation errors smaller than $\pm 1°$. With the goal of increasing the safety and robustness of the implemented algorithms, an orientation uncertainty range of $\pm 2°$ was considered (instead of the $\pm 1°$ measured in preliminary tests). By capping the orientation admissible range, it is possible to widen the surpassable translation errors of the developed methods (as seen in Figure 29).

Analyzing the proposed methods in the Subsection 4.4.2., it is evident that the proposed "Oscillation method complemented by orientation correction" is the only one not suitable for this application since its main focus is to surpass high orientation errors and small translation errors (which is the opposite of the requirements for this task). In fact, all other methods can be implemented directly since its admissible orientation error ranges are larger than the defined $\pm 2°$.

Table 10 shows the surpassable error ranges for all implemented methods, measured relative to the assembly workspace. These values differ from the ranges displayed in Table 9 as it is not necessary to overcome orientation errors as high as the previously developed ranges, therefore, it is possible to widen the surpassable translation errors of the developed methods (as seen in Figure 29).

Table 10. Implemented search methods surpassable error ranges in the assembly workspace

	x_A	y_A	α_A
Oscillation method	$[-10, 10]\ mm$	$[-10, 10]\ mm$	$[-4, 4]\ °$
Oscillation method complemented by environment exploration	$[-30, 30]\ mm$	$[-30, 30]\ mm$	$[-4, 4]\ °$
Environment exploration complemented by orientation correction	$[-60, 60]\ mm$	$[-60, 60]\ mm$	$[-4, 4]\ °$

In order to calculate the equivalent errors in the robot workspace, the system of equations (3) can be used. Table 11 lists the admissible translation error range in the robot workspace for different values of orientation error.

Table 11.Implemented search methods surpassable error ranges in the mobile robot workspace

	$\alpha = 0°$		$\alpha \in [-1, 1]\ °$		$\alpha \in [-2, 2]\ °$	
	x_R	y_R	x_R	y_R	x_R	y_R
Oscillation method	$[-10, 10]$	$[-10, 10]$	$[-1.3, 1.3]$	$[-9.9, 10]$	–	–
Oscillation method complemented by environment exploration	$[-30, 30]$	$[-30, 30]$	$[-21.3, 21.3]$	$[-29.9, 30]$	$[-12.6, 12.6]$	$[-29.7, 30]$
Environment exploration complemented by orientation correction	$[-60, 60]$	$[-60, 60]$	$[-51.3, 51.3]$	$[-59.9, 60]$	$[-42.6, 42.6]$	$[-59.7, 60]$

This table suggests that both the "Oscillation method complemented by environment exploration" and the "Environment exploration complemented by orientation correction" can be implemented successfully for the whole orientation error range of $\pm 2°$. However, the pure oscillation method can only be employed successfully for extremely low values of rotation errors. Since the value of error of $\pm 2°$ is an overestimation, and the preliminary tests suggested values of orientation error lower than $\pm 1°$, it is still possible that this method can be successfully implemented.

In fact, every actual test resulted in a successful implementation of all the methods, including the pure oscillation method. These results support the idea that the positioning system of the robot is accurate enough to guarantee an extremely low rotation error. Nevertheless, the oscillation method is being implemented on the limits of its capabilities, meaning that if abnormal factors cause the robot to approach the assembly position with slightly larger errors, this method will most likely fail.

This issue resulted in the identification of another crucial need for the implementation of these methods - a feedback tool capable of informing the system if the assembly was performed successfully or not. For this reason, at the end of each method, a boolean function is called, evaluating if the assembly was terminated with a collision detection condition and if the routine ended in the correct height (this condition can be implemented since the height of the whole system is constant).

This success test also allows the creation of another routine - the “Progressive method”. The goal of this method is to increase the robustness of the oscillation method. This routine performs the “Oscillation method” and, if success assessment determines that the assembly was failed, it calls the “Environment exploration complemented by orientation correction” method, which has a bigger range of admissible errors. Thus, in the eventuality that an abnormal positioning error caused the oscillation method to fail, a more forgiving method will be immediately called, ensuring that the assembly is successfully completed.

Finally, in order to prove the validity of the implemented strategies in an actual assembly task, the “Environment exploration complemented by orientation correction” method was calibrated to assembly the bearing cap in its real position for the assembly of the engine (Figure 36).

Figure 36. Successful assembly of the bearing cap in its actual working position

5.3. Communication with the mobile robot

In order to construct a fully autonomous system, the establishment of communication between the mobile robot and the manipulator's controller is crucial. Whenever the robot reaches the assembly station, it must be capable of communicating with the manipulator, instructing it to start one of the implemented assembly methods. In turn, the manipulator must be able to perform the solicited routine and inform the robot not only when it completed the routine, but also whether the assembly task was successful or not.

The mobile robot was implemented with Robot Operating System (ROS), a series of software libraries and tools that allow the implementation of robotic systems (from drivers, to control algorithms and developer tools). Its process management is achieved through the use of a task manager, which manages skills to call all the different operations and routines of the robot. As an example, this robot implements a "DriveSkill" to move the robot to a certain target position. Similarly, in order to command the start of an assembly routine, a skill was created to command the manipulator to start the selected method ("CommandSkill").

A communication solution for these types of solutions is the actionlib package of ROS which provides tools to create an action server and clients (Figure 37). Action servers receive a goal that, in turn, activates a certain routine. It can also send periodic feedback to the client and return a result at the end of the routine. The use of the actionlib package is especially beneficial for tasks with long executing times, as it has the ability to cancel the execution of a goal.

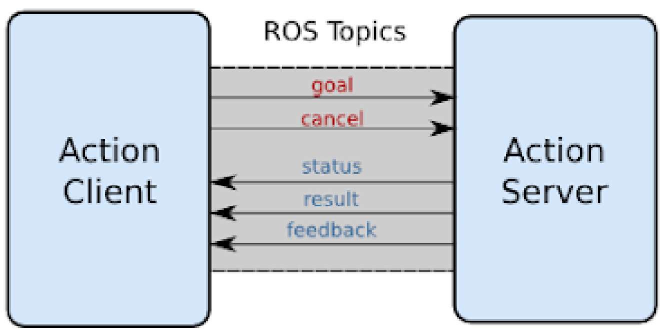


Figure 37. Communication between action server and client

Since the manipulator is controlled by a distinct controller, separate to the ROS workspace, an external action server running on the KUKA Sunrise Cabinet must be created. For that, a ROS action library for Java was used, "rosjava_actionlib". The necessary adjustments were made to the integrated action server so that necessary assembly routines could be called from within this server whenever the corresponding goal was received. Furthermore, the server is also responsible for notifying the action client on whether the assembly was successful or not, in the form of a result message.

The server was setup so that the system could solicit the manipulator to perform the "Oscillation method", the "Oscillation method complemented by environment exploration", the "Environment exploration complemented by orientation correction" method or the "Progressive method" introduced in Section 5.2., by sending a predetermined string variable within the goal message. This way, if the navigation system has the capability of estimating its positional uncertainties, it can call the method that best suit the needs for each individual scenario. Otherwise, if its positioning system cannot estimate its own error, it can call the progressive method, where the manipulator attempts increasingly flexible methods in order to accommodate every possible error value.

With this implementation, the robotic application running on the manipulator's controller is limited to initializing and continuously running the action server, where the actual assembly routines are called. Thus, this system architecture means that the mobile robot has full control of the system, i.e. it is responsible of performing trajectories, deciding which assembly routines to call and processing its result message, whereas the manipulator continuously runs the action server, waiting to receive a goal message.

The implementation of the different created skills is performed in the form of a state chart XML file, where a graphic interface allows an easy visualization of the created routine. Figure 38 displays an example of the state chart created to move the mobile robot to the assembly position ("DriveSkill"), perform the required assembly ("CommandSkill") and move to different positions depending on the result of the performed assembly.

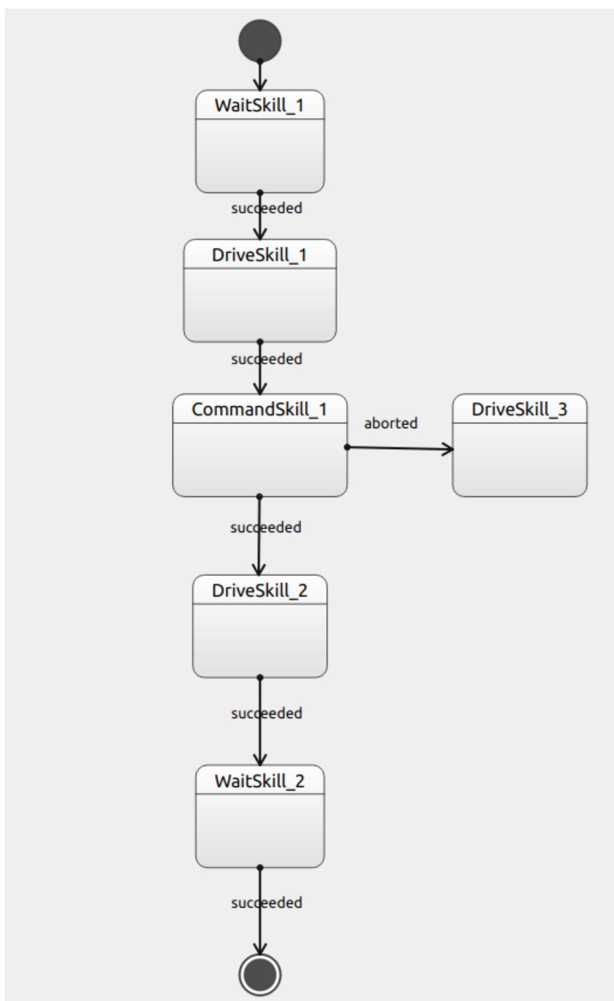


Figure 38. Example of the state chart file controlling skill sequence

6. Conclusions and future work

The present study aimed for the development of a mobile assembly station using force control as a mechanism to surpass positional uncertainties arising from the positioning of the mobile platform. Several different methods were developed and tested, showcasing the strengths and limitations of each one.

The contact force analysis method could not be implemented for the studied assembly task since the inherent force and torque measurements of the manipulator did not have enough accuracy to detect the needed low torque values caused by the required collisions. However, this strategy is still promising and, as such, it would be interesting to implement this method using a more accurate method of measuring the contact forces and torques (e.g. a dedicated sensor mounted on the robot's flange).

Multiple search methods were developed, using numerous compliant strategies to, ultimately, culminate on a successful assembly task. The pure oscillation method can be implemented directly to general assembly tasks (after the appropriate calibration of oscillation amplitude). However, the methods employing environment exploration and orientation correction using impedance control are highly reliant on the performed assembly task, as well as its parts geometry. Therefore, they are not easily adaptable for generic operations in other assembly stations. In fact, the implementation of these methods on other tasks would require the development of an identical method from scratch using similar strategies and would most likely result in different surpassable error ranges.

The search methods culminating with the oscillation strategy merely perform blind searches to insert a part into a predetermined slot. As such, since the assembly always has a certain tolerance, this method does not completely eliminate the positional uncertainties present on the assembly workspace. Thus, if multiple assembly tasks are to be performed in the same workspace, it is necessary to perform an oscillation routine for each one (even if the subsequent routines have smaller oscillation amplitudes than the first one due to the decreased positional uncertainties). Thus, methods using oscillation as the strategy to surpass the final positional errors tend to be more suitable for single assembly tasks on a given station.

On the other hand, the search method based solely on environment exploration and orientation correction determines the position of the workspace, fully eliminating the errors present on all degrees of freedom. As such, this routine is recommended for performing multiple assemblies in the same workspace since it only needs to run the search routine one time.

Aiming to overcome the limitations of the developed methods, a few tools were proposed as an attempt to broaden the scope of the developed methods to any general assembly task. An external accessory was proposed to optimize and implement the "Environment exploration complemented by orientation correction", making it compatible with generic assembly tasks with planar errors after a simple calibration process. The required length of this accessory is reliant on the required uncertainties ranges; however, since oversized accessories do not negatively impact the performance of the routine, it is preferable to slightly oversize it. Furthermore, since the optimal design solution of this accessory is extremely simple, it can be easily improvised and implemented.

In turn, the proposed CAD analysis algorithms can be used to automatically identify potential locations within the assembly station where reachable orthogonal surfaces can be used for the implementation of the proposed search method. With this approach, it is not necessary to introduce extra components in the assembly station, but its implementation does not ensure the success of any general assembly because it is still reliant on the geometry of the present components.

Additionally, it was concluded that the geometry of the gripped workpiece is essential for successful orientation using impedance control, since it must have at least one flat surface. Thus, it is not possible to implement the developed method with a gripped cylindrical workpiece. However, since the proposed routine fully eliminates the present errors, this limitation can be overcome by running the routine with an appropriate workpiece, followed by the direct assembly of the cylindrical part. Furthermore, in order to increase the flexibility of the system, the mobile platform can carry a generic prismatic component to successfully perform the correction algorithm even if none of the assembled parts have a flat surface.

The establishment of the correlation between the assembly and the mobile robot workspaces revealed that for these situations, orientation errors of the mobile platform are crucial since they are translated to further translation errors in the assembly workspace. Therefore, the surpassable error range on the assembly workspace will always be larger than the equivalent range on the mobile platform workspace. Since this effect is amplified by the distance between the manipulators base and the assembly position, the bigger this distance is, the larger this error translation will be and the more important it is to limit the orientation errors of the platform.

The implementation of the developed search methods on the final mobile robot setup revealed to be quite straightforward, only requiring simple calibrations of reference frames and force conditions values (which were only needed because the estimation of the external forces is reliant on the joints configuration at that point). Since the actually implemented orientation errors were smaller than the ones originally defined on the developed setup, the methods could be implemented for larger translational error ranges.

The evaluation of the assembly success was crucial as it is a natural requirement for the development of a robust mobile assembly station. Furthermore, it allowed the creation of a “Progressive method”, which increased the robustness of the oscillation method by calling a more forgiving method when it detected that the “Oscillation method” failed.

Finally, in order to fully prove the validity of the implemented strategies in real assembly tasks, the actual position of the bearing cap within the motor was calibrated and the developed methods were able to easily accomplish this task.

References

[1] D. Gravel, F. Maslar, G. Zhang, S. Nidamarthi, H. Chen, and T. Fuhlbrigge, “Toward robotizing powertrain assembly,” in *2008 7th World Congress on Intelligent Control and Automation*, 2008, pp. 541–546.

[2] J. Marvel, J. Falco, and J. Marvel, *Best practices and performance metrics using force control for robotic assembly*. US Department of Commerce, National Institute of Standards and Technology, 2012.

[3] D. Erickson, M. Weber, and I. Sharf, “Contact stiffness and damping estimation for robotic systems,” *Int. J. Rob. Res.*, vol. 22, no. 1, pp. 41–57, 2003.

[4] A. Kramberger, A. Gams, B. Nemec, D. Chrysostomou, O. Madsen, and A. Ude, “Generalization of orientation trajectories and force-torque profiles for robotic assembly,” *Rob. Auton. Syst.*, vol. 98, pp. 333–346, 2017.

[5] L. Roveda, G. Pallucca, N. Pedrocchi, F. Braghin, and L. M. Tosatti, “Iterative learning procedure with reinforcement for high-accuracy force tracking in robotized tasks,” *IEEE Trans. Ind. Informatics*, vol. 14, no. 4, pp. 1753–1763, 2017.

[6] S. P. Chan and H. C. Liaw, “Generalized impedance control of robot for assembly tasks requiring compliant manipulation,” *IEEE Trans. Ind. Electron.*, vol. 43, no. 4, pp. 453–461, 1996.

[7] J. A. Marvel, W. S. Newman, D. P. Gravel, G. Zhang, J. Wang, and T. Fuhlbrigge, “Automated learning for parameter optimization of robotic assembly tasks utilizing genetic algorithms,” in *2008 IEEE International Conference on Robotics and Biomimetics*, 2009, pp. 179–184.

[8] J. Wang, G. Zhang, H. Zhang, and T. Fuhlbrigge, “Force control technologies for new robotic applications,” in *2008 IEEE International Conference on Technologies for Practical Robot Applications*, 2008, pp. 143–149.

[9] J. Bös, A. Wahrburg, and K. D. Listmann, “Iteratively Learned and Temporally Scaled Force Control with application to robotic assembly in unstructured environments,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3000–3007.

[10] D. P. Gravel and W. S. Newman, “Flexible robotic assembly efforts at Ford Motor Company,” in *Proceeding of the 2001 IEEE International Symposium on Intelligent Control (ISIC’01)(Cat. No. 01CH37206)*, 2001, pp. 173–182.

[11] F. Suárez-Ruiz and Q.-C. Pham, “A framework for fine robotic assembly,” in *2016 IEEE international conference on robotics and automation (ICRA)*, 2016, pp. 421–426.

[12] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, “Force controlled robotic assembly without a force sensor,” in *2012 IEEE International Conference on Robotics and Automation*, 2012, pp. 1538–1543.

[13] A. Stolt, M. Linderoth, A. Robertsson, and R. Johansson, “Detection of contact force transients in robotic assembly,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 962–968.

[14] E. A. Erlbacher, “Force control basics,” *Ind. Robot An Int. J.*, 2000.

[15] B.-H. Yang and H. Asada, “Learning compliance control laws for robotic assembly: an application of

adaptive reinforcement learning control," in *Proceedings of 1993 International Conference on Neural Networks (IJCNN-93-Nagoya, Japan)*, 1993, vol. 2, pp. 1821–1824.

[16] H. Chen, J. Wang, G. Zhang, T. Fuhlbrigge, and S. Kock, "High-precision assembly automation based on robot compliance," *Int. J. Adv. Manuf. Technol.*, vol. 45, no. 9–10, p. 999, 2009.

[17] Odham, "Active and Passive Force Control Robotic Deburring - A Complete Picture," in *Application Engineer, ATI Industrial Automation—September 2007*, 2007.

[18] F. Zeng, J. Xiao, and H. Liu, "Force/torque sensorless compliant control strategy for assembly tasks using a 6-DOF collaborative robot," *IEEE Access*, vol. 7, pp. 108795–108805, 2019.

[19] J. De Schutter, J. Katupitiya, P. Vanherck, and H. Van Brussel, "Active force feedback in industrial robotic assembly: a case study," *Int. J. Adv. Manuf. Technol.*, vol. 2, no. 4, pp. 27–40, 1987.

[20] B. Komati, M. R. Pac, I. Ranatunga, C. Clévy, D. O. Popa, and P. Lutz, "Explicit force control vs impedance control for micromanipulation," in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013.

[21] C.-C. Cheah and D. Wang, "Learning impedance control for robotic manipulators," *IEEE Trans. Robot. Autom.*, vol. 14, no. 3, pp. 452–465, 1998.

[22] H. Chen and Y. Liu, "Robotic assembly automation using robust compliant control," *Robot. Comput. Integr. Manuf.*, vol. 29, no. 2, pp. 293–300, 2013.

[23] P. R. Giordano, A. Stemmer, K. Arbter, and A. Albu-Schaffer, "Robotic assembly of complex planar parts: An experimental evaluation," in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3775–3782.

[24] KUKA Roboter GmbH, "Operating and Programming Instructions for System Integrators." 2017.

[25] M. P. Polverini, A. M. Zanchettin, and P. Rocco, "A constraint-based programming approach for robotic assembly skills implementation," *Robot. Comput. Integr. Manuf.*, vol. 59, pp. 69–81, 2019.

[26] KUKA Roboter GmbH, "Sensitive robotics - LBR iiwa." 2017.

[27] Robotiq Inc, "2F-85 & 2F-140 - Instruction Manual." 2020.