Redesigning Chemistry Laboratories: Accelerating Material Discovery through Simulation and Cutting-Edge Robotics

Master Thesis

Group 1054 / ROB10

Aalborg University
Electronics and IT

Electronics and IT
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Redesigning Chemistry Laboratories: Accelerating Material Discovery through Simulation and Cutting-Edge Robotic

Theme:
Master Thesis

Project Period:
Spring Semester 2023

Project Group:
Group 1054

Participant(s):
Andrei Voica
Daniel Moreno París

Supervisor(s):
Simon Bøgh

Copies: 0

Page Numbers: 85

Date of Completion:
June 1, 2023

Abstract:

This project focuses on redesigning the chemistry laboratories into self-driven laboratories capable of accelerating the new materials discovery task. By using robotic solutions and simulation, the project aims at optimising the chemical processes that are mainly done by humans, which can be slow and error-prone. The self-driven laboratory will be implemented on the AAU Matrix Production setup that consists of 5 Kuka robotic manipulators, the B&R Automation Acopos 6D magnetic levitation platform and different purpose-made parts. For development purposes, Nvidia Isaac Sim is used where the physical setup will be replicated in a simulated environment where different experiments can be carried out. Robot Operating System1 (ROS1) is used to control the simulated Kuka manipulators as well as the real ones. The simulation experiments show that the system is capable of automatically completing a chemical process, but transferring it to the physical setup proved to be a challenging task.

Contents

Abbreviations

AEP	Automous Experimental Platform
API	Application Programming Interface
BNN	Bayesian Neural Network
CAD	Computer-Aided Design
CFR	Continuous Flow Reactors
DDS	Data Distribution Service
GUI	Graphical User Interface
HTE	High-Throughput Experimentation
LED	Light-Emitting Diode
MAP	Material Acceleration Platform
ML	Machine Learning
MSA	MoveIt Setup Assistant
NLP	Natural Language Processing
OSL	Organic Semiconductor Laser
OPV	Organic Photovoltaics
PMC	Planar Motor Controller
PtX	Power-to-X
QD	Quantum Dots
QLED	Quantom Dot LED
ROS	Robot Operating System
UI	User Interface
URDF	Universal Robot Description Format
USD	Universal Scene Description
VR	Virtual Reality
RSI	Robot Sensor Interface
KSS	Kuka Software System
OLP	Off-Line Programming
SDF	Signed Distance Field

Preface

This master thesis is made by Group 1054, on the 10^{th} semester of Robotics, under the Department of Electronics and IT at Aalborg University. The project was made over the course of four months starting at the beginning of February and was supervised by Simon Bøgh.
The Github repository containing the implementation can be accessed using the following link. Some examples of what has been achieved are shown in the following Youtube link.

Aalborg University, June 1, 2023

Andrei Voica
<avoica18@student.aau.dk>

Daniel Moreno París
<dmoren21@student.aau.dk>

Chapter 1

Introduction

In the historical context of the design of chemistry laboratories, it is evident that their conception is rooted in an era dominated by manual labour [1]. The rigidity of these century-old design paradigms poses significant challenges to the implementation of any form of automation, notably, the integration of robotic manipulators.

Nowadays, a signigicant amount of research is focused on mitigating climate change and its consequences, hence, cutting-edge technology has been used to address this problem. New technological advances such as Material Acceleration Platforms (MAPs) are emerging to accelerate the discovery of new materials as this can potentially contribute to reducing the impact of climate change. These technologies incorporate Artificial intelligence, robotic systems, orchestration, databases and human insight, working within a closed-loop framework. [2]

1.1 Motivation

MAPs are a new paradigm in materials science that aims to reduce the time and cost of discovering and optimising new materials. By discovering materials with desirable properties for energy production, conversion, storage and efficiency, MAPs have the potential to solve climate change-related challenges. However, it is not limited to Power-to-X challenges, since this technology aims to significantly accelerate the discovery of new materials, in a nearly infinite study space, it brings a whole new world of possibilities for a wide range of application fields.

Another strength of MAPs over traditional synthesis methods is that they allow a more efficient and sustainable approach by reducing waste, energy consumption, and environmental impact. Furthermore, being an automated process, data acquisition can be done automatically in a standardised format facilitating collaboration among researchers.

The potential of MAPS is enormous, but so are the challenges that must be overcome to make it a practical reality beyond the field of research. Some of the challenges that MAPs face are ensuring data quality, reliability, and reproducibility; integrating different sources of data and knowledge; dealing with uncertainty and complexity; and ensuring ethical and social responsibility. [3]

As we have had the opportunity to see on several visits to the organic chemistry laboratories at Aalborg University, they are entirely tailored for manual use by human researchers. These laboratories are very complex and chaotic to allow the use of robots in a useful way in this context. Because of this situation, in order to accelerate the speed at which new materials synthesis experiments are carried out, a completely different approach has to be sought that allows parallelisation of experiments, quality data collection, as well as some flexibility to recreate new experiments without the need to adapt the entire laboratory.

The research with MAPs is a long-term project at Aalborg University in which we will develop the simulation part of the physical setup. As this project will be developed over several years, an essential requirement is that we use Nvidia Isaac Sim to develop the simulation environment so that our work is usable for the project and future research. Furthermore, Aalborg University is collaborating with the Pioneer Centre CAPeX to develop a MAP [4]. The infrastructure aims to enable direct discovery, synthesis and design of new catalysts and other materials targeted to their specific operating conditions, which are necessary for Power-to-X applications.

Our motivation in this project is to find effective ways to leverage the potential of robotics and artificial intelligence to drive the processes involved in the discovery of high-performance materials for clean energy technologies, among other fields, that have the potential to drive the transition to a low-carbon future.

1.2 Initial Problem Formulation

Being aware of the potential of robotic automation and considering the challenge of integration in the human-dominated field of materials science, where the vast majority of processes require the versatility and knowledge of subject matter experts, leads to the initial formulation of the problem:

How can a robotic platform help in the discovery of new materials in a flexible and efficient way?

Chapter 2

Problem Analysis

In this chapter, an exploration of the potential domains of the initial problem formulation is carried out by conducting a comprehensive review of the various state-of-the-art techniques and algorithms for MAPs. This is followed by a brief overview of the basic theory.

2.1 Material Acceleration Platforms

Material Acceleration Platforms (MAPs) are systems that use high-performance computing, artificial intelligence and robotic systems to speed up the discovery of new materials by conducting experiments autonomously. Additionally, it is important to include human intuition in this process since, in the field of chemistry, there is a lot of empirical knowledge that cannot be found in current databases or is poorly documented. Figure 2.1 shows the sub-fields that MAPs encompass in pursuit of autonomous experimentation. [5]

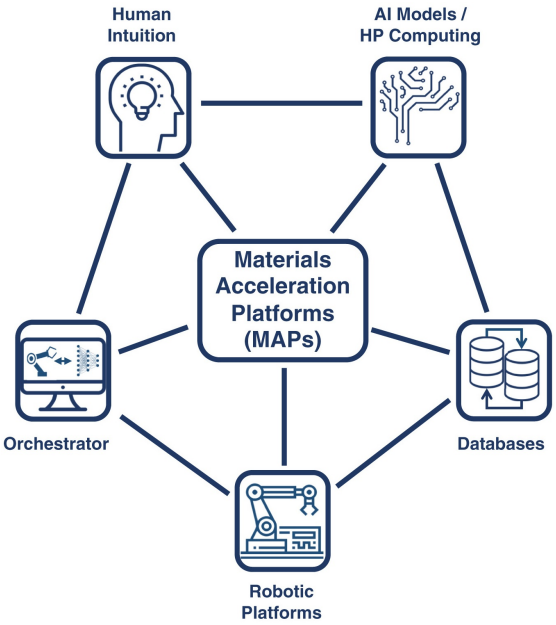

Figure 2.1: The main areas that Material Acceleration Platforms involves are human intuition, AI models with high-performance computing, high-quality databases, robotic platforms for automated experiments and orchestrator software for communication between modules, by Flores-Leonar et al. Reprinted from [5] Copyright 2020, with permission from Elsevier.

In general, MAPs consist of a closed loop, as shown in Figure 2.3, which includes four phases. A first phase where artificial intelligence is used to design experiments according to target objectives (e.g. properties, price) and constraints (e.g. inventory, stability). In a second phase, robotic systems are used to carry out the designed experiment in an automated way. Once the material has been synthesised, an automatic characterisation process is carried out where the properties of the material are tested. Finally, an analysis of the results obtained is carried out, leveraging the data obtained to update the database and improve the design of future experiments.

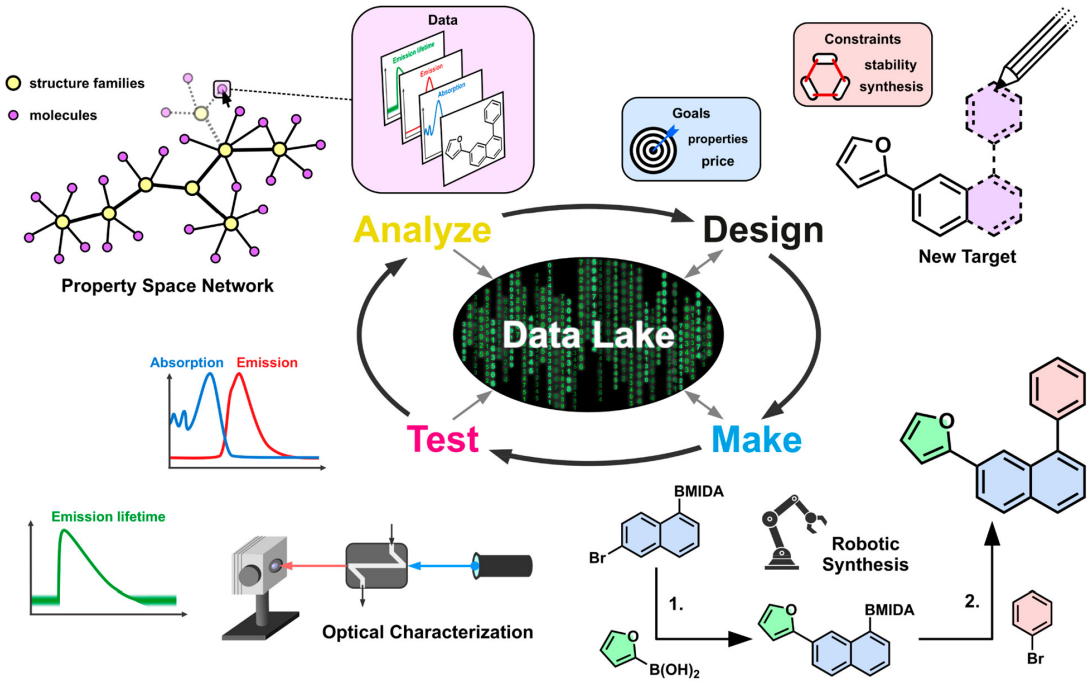

Figure 2.2: Schematic of MAPs cycle closed-loop (Design, Make, Test, Analyze). Reprinted with permission from [3]. Copyright 2022 American Chemical Society.

The design of experiments is a key part of the cycle. Since the exploration space is humongous and high-dimensional, it remains infeasible and extremely impractical to enumerate all combinations. This approach would generate too vast amounts of data to process in order to establish the synthesis-structure-property relationship. Consequently, intelligent decision-making and data analysis algorithms are essential for conducting autonomous experiments, constituting the basis for developing Material Acceleration Platforms. [6]

The synthesis of new materials is a process that involves many steps and the techniques applied to the chemical agents vary enormously depending on the desired target. Traditionally, the synthesis process has been performed manually by experts in the field.

On the one hand, human cognitive abilities are often able to perform these tasks with ease, relying on visual feedback and on-the-fly problem solving. On the other hand,

automating these processes can bring many benefits such as increased reproducibility and precision, enhanced data collection, allows parallelisation of experiments, avoid human error and reducing human exposure to hazardous substances, albeit is very challenging since, until now, laboratories and the required machines have been developed primarily for human workers. For these reasons, MAPs aim to completely redesign the materials synthesis workflow to reduce the bottlenecks of traditional experimentation and take advantage of all the advantages that robotic automation provides. [3]

Once the synthesis of the material has been carried out, it is equally important to characterize the properties of the material, i.e. analyzing and understanding the properties of materials being synthesised or tested. Characterization includes techniques such as spectroscopy, chromatography and other analytical methods. Similar to material synthesis, this process is generally carried out manually by human experts, and the same advantages are to be found in its automation, including the documentation of failed processes that is so lacking in the scientific literature of the field. As discussed in the article published by Christensen et al. [7], the biggest challenge at this stage is that manufacturers hardly produce machines adapted for integration (both physically and software-wise) with robots, which adds extra difficulty to the tasks performed by the robots and requires considerable time investment to write custom code, but it seems likely that this problem will be solved in the coming years as laboratories gradually move towards automation.

After obtaining the data resulting from the characterization process, an analysis is required. The raw data must be interpreted to obtain quantitative values for the properties of the synthesised materials. Traditionally, subject matter expert interpretation has been used to assess the results, however, similarly to the other processes in the cycle, MAPs aim to automate this process and exploit the data gathered to optimise subsequent experiments. One of the most relevant challenges in this stage is the automated identification of unknown compounds when the result is far from what is expected, as reported in Blazenovic et al. [8].

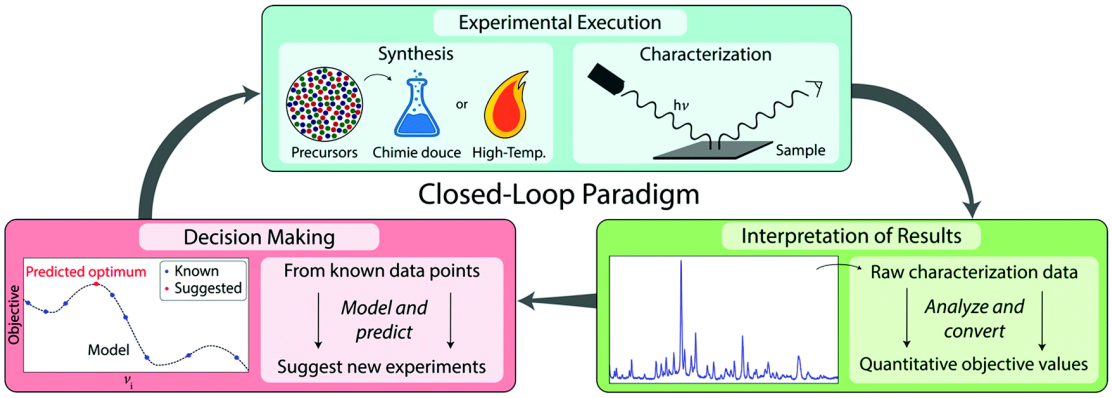

Figure 2.3: Closed loop paradigm in MAPs by Szymanski et al. from Ref. [9], reproduced under the license ID: 1360997-1

2.2 State of the art

This section presents different approaches towards the accelerated discovery of new materials proposed by several research groups. Moreover, different tools that help to enable the acceleration towards the discovery of new materials will be presented.

2.2.1 Self-Driving Laboratories

A self-driving laboratory can be referred to as a system that is able conduct a chemical experiment without the intervention of a human. These system can be developed using different tools and automation approaches.

Continuous flow reactors (CFR) were developed with the intention of replacing the traditional reactors, such as flasks and beakers. The limitations of the traditional reactors are that they suffer significant batch-to-batch variability and can generate unexpected byproducts, impeding the accurate investigation of different reactions were the repeatability and controlled reactions are of high importance. The advantages of using CFR are that they enable the possibility of using smaller quantities of liquid, ranging from microliters to nanoliters, better heat transfer due to the small diameter of the channels, increase the repeatability of the experiments and the ability to modularize the setup. [6]

Artificial chemist

Epps et al. [10] created a system using CFR where they synthesize colloidal QDs. The CFR approach allows them to greatly reduce the time and the cost of the batch techniques. The setup contains a precursor formulation module that consists of multiple syringes, a flow reactor module and an in situ material characterization module.

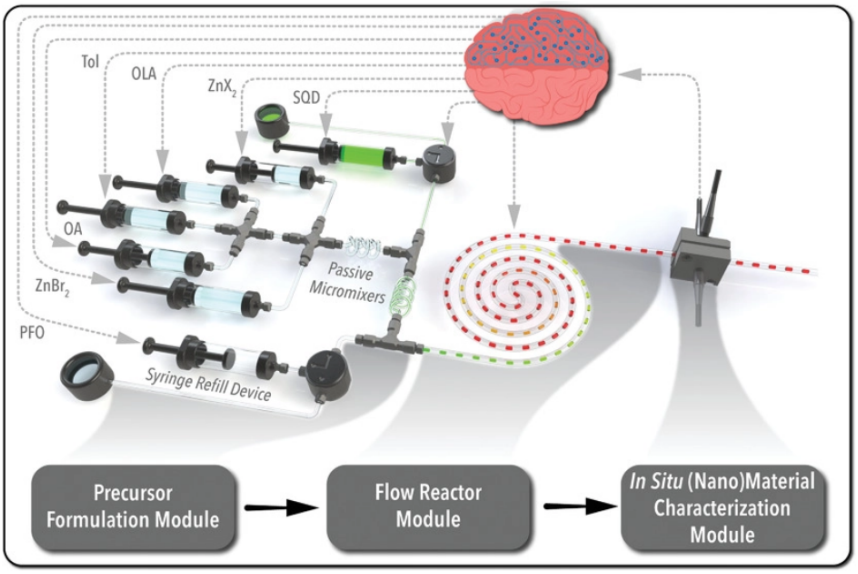

Figure 2.4: Schematic of the Artificial chemist containing the Precursor Formulation Module, the Flow Reactor Module and the *In Situ* Characterization Module. Reproduced with the permission from [10].

Desktop robots were designed to overcome the CFR's main limitation, which is the lack of agility and the lack of conducting different chemical experiments. [6] Different researchers were able to design their own solutions that would solve the lack of agility problem posed by CFRs.

Chemspeed

Chemspeed is a commercially available solution for automated process in chemistry. One of the main feature is the capability of parallel workflow allowing for organic, ligand, organometallic, and nanomaterials synthesis. Alongside the parallel workflow, Chemspeed is able to dispense solids and liquids with a resolution in micro grams, can reach temperatures between -70°C and 200°C, it has different analytical tools integrated, such as HPLC, and the software has a simple user interface with many drag & drop modules. [11]

Figure 2.5: Representation of the Chemspeed's commercial solution [11].

Ada

MacLeod et al. [12] developed a modular and flexible self-driving laboratory called "Ada" that is capable of synthesizing, processing and characterizing, in an autonomous way, organic thin films. The system has a robot that is equipped with a pipette that is connected to a syringe pump and a pneumatic gripper for substrate handler, a spin coater able to achieve 1000 rpm, an annealing oven that heats up the material to 165°C, a camera, a UV-vis-NIR spectrometer and a 4-point probe for analyzing the conductivity of the film. Moreover, Ada is able to learn and design experiments by itself using a global Bayesian optimization.

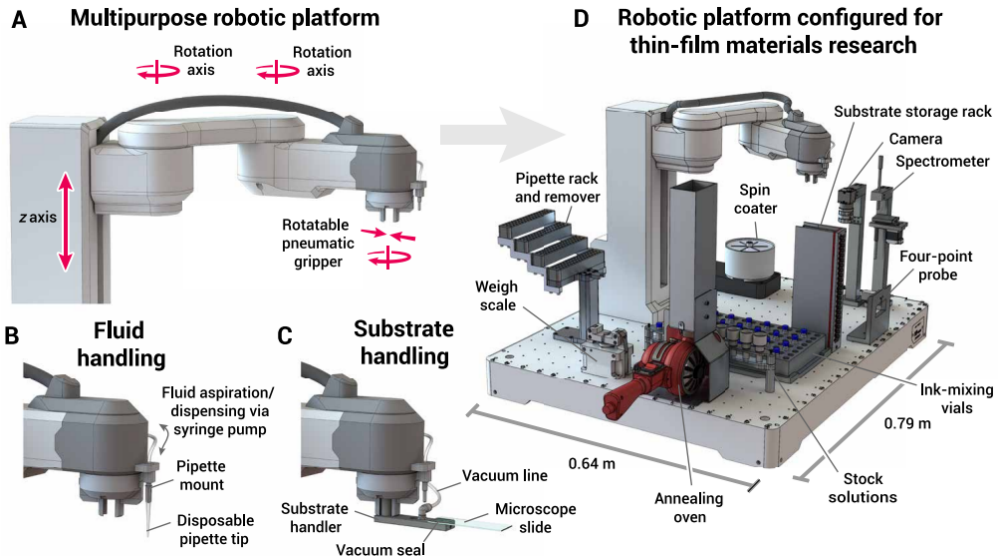

Figure 2.6: Representation of Ada, the self-driving laboratory for fabrication and characterization of thin-film materials. Reproduced under a Creative Commons Attribution NonCommercial License 4.0 (CC BY-NC). from Ref. [12], Copyright 2020 AAAS.

Ada has a 8 step autonomous workflow for conducting the experiments. The system starts by preparing the precursor solutions that are after send to the spin coater. Once the solution was spread, it is send to the annealing heater where it is heated to 165°C. Once the heating process is done, the material is now brought back to the ambient temperature and a picture is taken to check for any defects. After the picture is taken, the optical spectra is characterized using UV-vis-NIR. Next step is to determine the conductivity of the film, and it is done using a 4-point probe. The last 2 steps of the process are to compute a pseudomobility based on the conductivity and the spectra of the film and to determine the next experiment. [12]

WANDA

Chan et al. [13] created an automated synthesis platform for colloidal inorganic nanocrystals, called WANDA (Workstation for Automated Nanomaterials Discovery and Analysis). WANDA consists of a liquid handling robot, a heated needle, a vial-gripper for handling and an automated balance for recording the masses. Because the required temperatures can reach up to around 300°C, they also created a custom deck element containing eight high temperature reactors.

The motivation behind this system is to enable the easier optimization of these nanomaterials properties since the task can be overwhelming due to the large number of variables that have to be considered for nanocrystal growth. Moreover, through automation, the system can precisely control the reaction parameters and offer a great repeatability. [13]

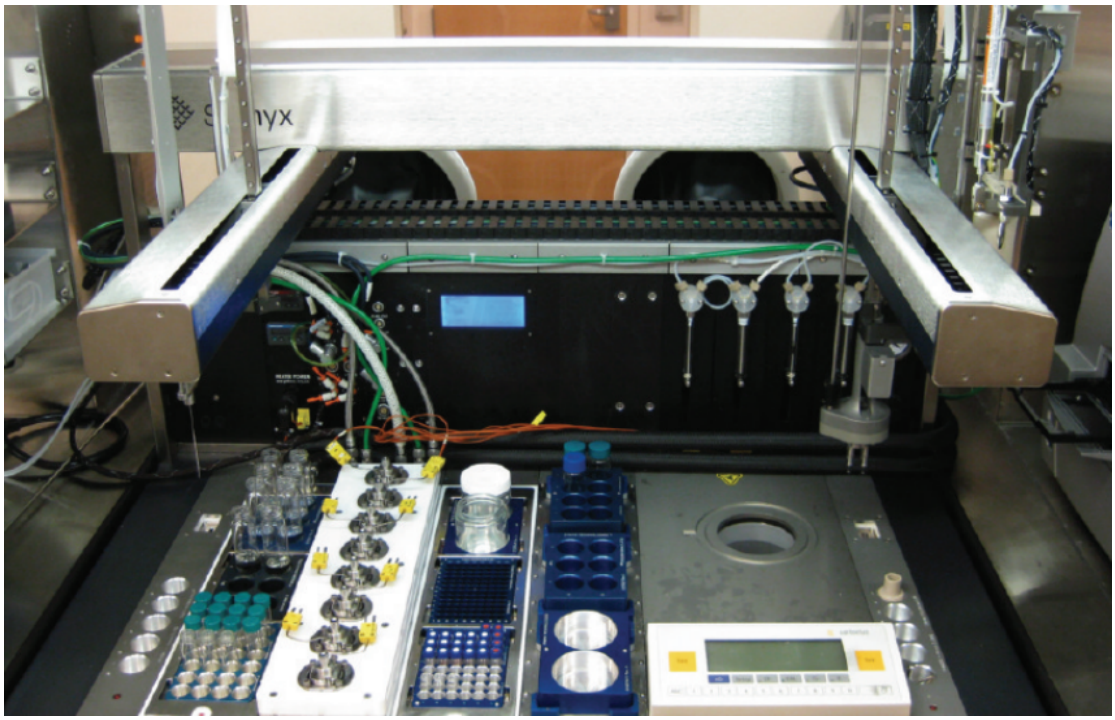

Figure 2.7: Close view of the Wanda system and setup. Reprinted with permission from Chan et al. [13]. Copyright 2023 American Chemical Society.

AMANDA - LineOne (L1)

Wagner et al. [14] have designed a platform capable of controlling multiple MAPs called AMANDA (Autonomous Materials and Device Application). The main focus on the work conducted by Wagner et al. [14] is the acceleration of thin-film material discovery part of the system, called LineOne(L1), using automated devices orchestrated by their own software, AMANDA software, that is capable of autonomously synthetizing and characterizing the process. The whole process for thin-film materials is similar to the "Ada" system developed by MacLeod et al. [12].

Their approach to implementing AMANDA software provides the ability to control multiple MAPs. Wagner et al. [14] designed the software to be highly flexible and support a wide range of equipment, with a fast integration. Moreover, the software allows for operating on sequence plans that can be chosen arbitrarily or generated automatically, based on the needs of the research facility. On Figure 2.8 AMANDA L1 is shown.

Figure 2.8: AMANDA LineOne(L1) on the left and the control center on the right. Reproduced under a Creative Commons Attribution 4.0 International (CC BY 4.0). from Ref. [14], Copyright 2020 AAAS.

Dropfactory

Grizou et al. [15] developed a robotic solution, called Dropfactory, capable of conducting experiments on dynamic oil-in-water droplets where they are testing their search algorithm, the workflow implemented can be seen in Figure 2.9. The motivation behind using oil-in-water droplets is because such a system poses many challenges due to the complex and poorly understood system and because a few components can create a wide range of behaviours and properties, that can be relevant in different industries. The platform can conduct more than 30 experiments per hour [15].

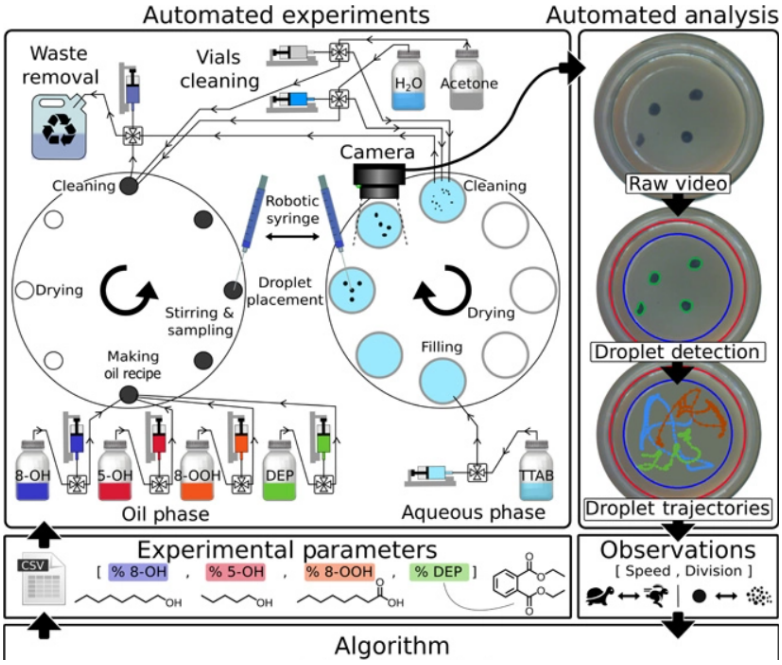

Figure 2.9: Schematic and workflow of the robotic solution Dropfactory. Reproduced under a Creative Commons Attribution License 4.0 (CC BY) from [15].

While the desktop robots are able to increase the agility of the system, a new limitation arises, which is that some characterization tools are big and complex and cannot be seamlessly integrated into the workflow of the desktop robot. In most cases, the characterization tool requires a special working environment, away from the desktop robot. This transfer would normally require human intervention, but there are some proposals that are trying to implement mobile robots into the workflow.

Mobile chemist

Burger et al. [16] developed a mobile robot able to interact in a conventional and unmodified laboratory due to its human-like size and reach. The robot is equipped with laser scanner and touch feedback, allowing it to operate in complete darkness which can be beneficial for light sensitive experiments. Another great advantage of this mobile platform is that it complies with the safety standards for collaborative robots allowing researchers and laboratory technicians to work along side it in the same environment.

Figure 2.10: The Kuka mobile chemist platform. Reproduced from Burger et al [16], with permission from Springer Nature.

The Table 2.1 highlights the most important self-driving laboratories in the field of materials acceleration discovery.

Laboratory Name	Author	Application
Artificial Chemist	Epps et al. [10]	Continuous Flow Reactor
Chemspeed	Cheemspeed technologies [11]	Desktop Robots
Ada	Macleod et al. [12]	Desktop Robots
WANDA	Chan et al. [13]	Desktop Robots
AMANDA L1	Wagner et al. [14]	Desktop Robots
Dropfactory	Grizou et al. [15]	Desktop Robots
Mobile Chemist	Burger et al. [16]	Mobile Robots

Table 2.1: Related work on self-driving laboratories with their main application.

2.2.2 Orchestrators

An orchestrator is a software framework that coordinates and manages the execution of automated experiments. It is doing so by selecting different functions that are based on the output of the previous function. The entire goal of an orchestrator is to complete an experiment from start to end. Since different chemical processes need different steps, an orchestrator will send commands to the system based on the needs of the said chemical process.

ChemOS

ChemOS is an orchestration software platform for autonomous experimentation in chemistry and materials science developed by Roch et al. [17], [18]. It uses ML algorithms to empower scientists to optimise and accelerate their research by automating the design, execution and analysis of experiments.

ChemOS enables researchers to explore large experimental spaces quickly and efficiently, with far greater potential than traditional trial-and-error methods. This orchestrator software provides the necessary structure to implement and operate self-driving laboratories. ChemOS is available in a modular and portable package for easy integration with automated equipment, in addition, allows remote control of laboratories. Finally, ChemOS allows fully autonomous experiments, as well as input and feedback from scientists as part of an experimental loop through an NLP module that categorises incoming messages as requests or feedback.

ESCALATE

ESCALATE is an open-source orchestrator software presented by Pendleton et al. [19] that stands for *Experiment Specification, Capture and Laboratory Automation Technology*. It uses an ontological framework to design machine-readable experiments to create perovskite crystals. ESCALATE was designed to simplify the data gathering process by providing an abstraction layer for human interaction and enables the integration of ML algorithms. In addition, the orchestrator is able to automatically generate a report from the available data.

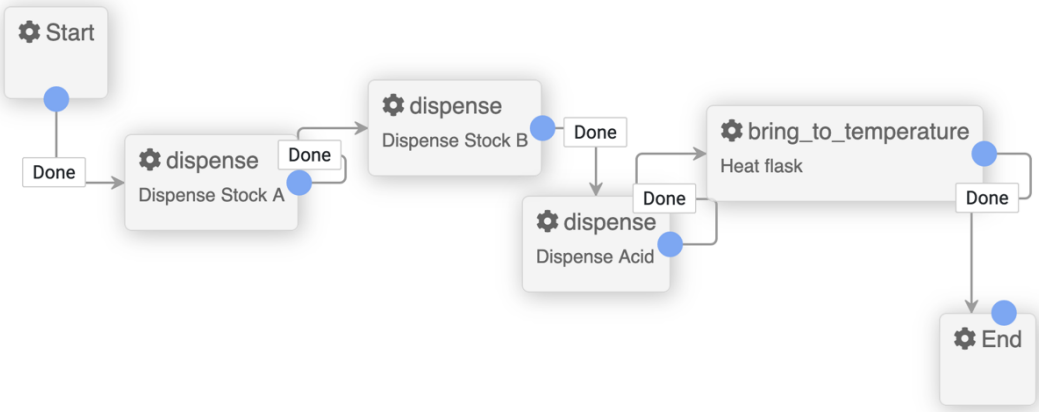

Figure 2.11: Example of experiment workflow created by ESCALATE orchestrator.

MAOS

The Materials Acceleration Operation System (MAOS), developed by Li et al. [20], is an orchestrator with the particularity that integrates virtual reality (VR). It features a replica of the real laboratory, with a customised user interface overlay, for human-robot interaction as shown in Figure 2.12. In this case, researchers utilised MAOS for the synthesis of cadmium (Cd) and selenium (Se) (CdSe) quantum dots used to absorb and emit visible radiation [21].

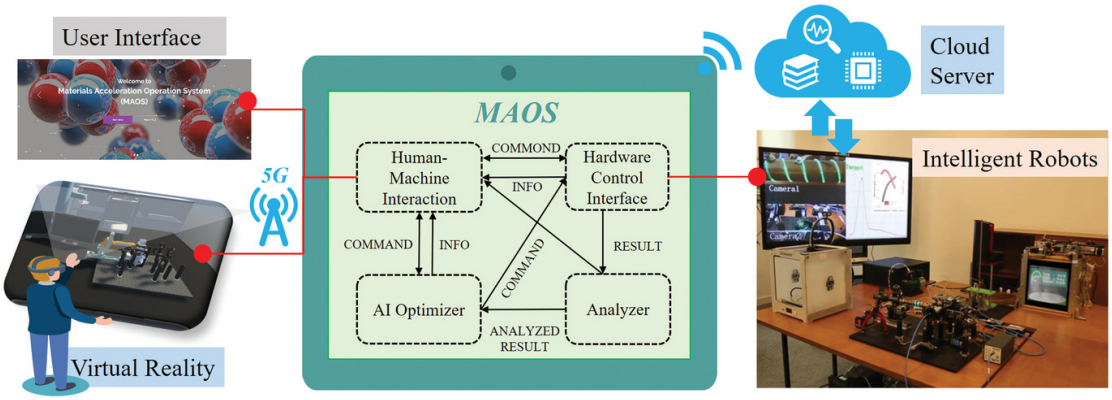

Figure 2.12: Schematics of MAOS system, which comprises integrated modules such as Human–machine interaction, hardware control interface, AI optimizer and analyzer. Users can engage with the robots using the website user interface (UI) and virtual reality, while the cloud server offers storage, computing, and searching services for the system by Li et al. Reproduced under a Creative Commons Attribution License 4.0 (CC BY 4.0). from Ref. [20], Copyright 2020 AAAS.

RXN

RXN for Chemistry, developed by IBM, is an AI-based tool provided with online access that is capable to predict the results of chemical reactions and optimize synthesis methods, as well as to generate chemical procedures automatically for use in both manual and automated laboratory operations. Features include reaction prediction [22], [23], retrosynthesis prediction [24], text to experimental procedure [25] and text-based chemical reactions to experimental procedure representation [26].

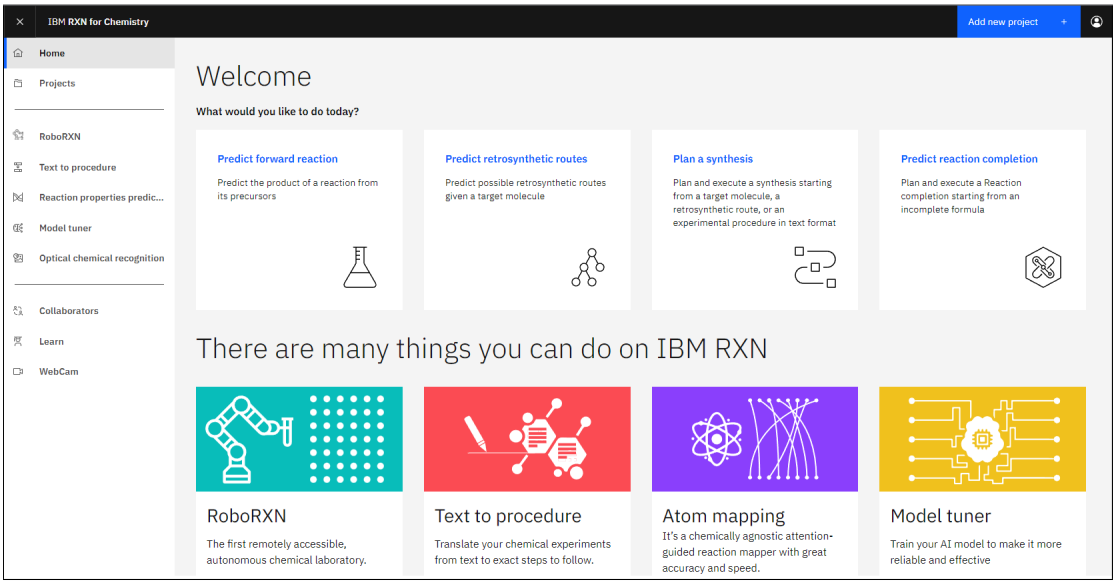

Figure 2.13: RXN for Chemistry main functionalities. Accessed: 15/03/2023, [27].

Table 2.2 provides an overview of the most relevant orchestrators currently available in the field of self-driving laboratories. In addition, the most important features are briefly outlined.

Orchestrator	Author	Special Features
ChemOS	Roch et al. [17], [18]	Variety of optimizers implemented, modular development, remote control of laboratories and human feedback supported
ESCALATE	Pendleton et al. [19]	Open-source, machine-readable experiments, ML algorithms integration and report generation
MAOS	Li et al. [20]	VR for human-robot interaction (UI overlay and laboratory replica) and AI optimizer and analyzer
RXN	IBM [27]	Reaction and retrosynthesis prediction, text to experimental procedure, remote control of laboratories, AI model tuning and web interface

Table 2.2: Related work on self-driving laboratories orchestrators

2.2.3 Planning Algorithms

In the MAP applications, a planning algorithm is used to choose the most optimal experiment towards the discovery of a new material. Such algorithms can take into consideration different metrics from past experiments and, based on the available data, choose a new experiment with slightly different variables that is believed to yield a better result.

Phoenics

The Phoenics algorithm, as proposed by Häse et al. [28], introduces a significant advancement in the optimization of experimental and computational processes. By employing Bayesian optimization, it systematically identifies the set of conditions necessary to meet specific objectives. This approach leverages a probabilistic model to predict the outcome of unobserved experiments, quantifying uncertainty and making informed decisions about the next steps to take. The algorithm's workflow, depicted in Figure 2.14, illustrates its sequential operation.

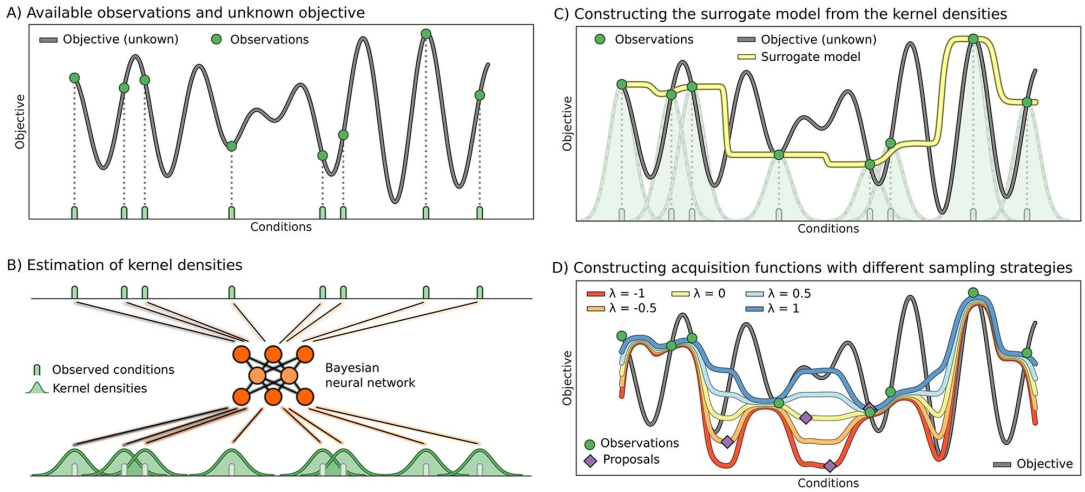

Figure 2.14: Phoenics algorithm workflow. (A) Unknown objective function resulting from an experimental procedure or computation that may have a high-dimensional nature. In this illustration, the objective function has been evaluated at eight different conditions (shown in green). (B) By using a Bayesian neural network, the observed conditions are analysed to generate a probabilistic model for estimating parameter kernel densities. (C) The surrogate model is created by assigning weights to the estimated parameter kernel densities based on their corresponding observed objective values. (D) By adjusting a single sampling parameter λ, the surrogate can be reshaped globally to prioritize either exploration (shown in red) or exploitation (shown in blue) of the parameter space. by Häse et al. Reprinted with permission from ref [28]. Copyright 2018 American Chemical Society.

Gryffin

Häse et al. [29] developed a global optimisation strategy, named Gryffin, implementing a Bayesian optimisation framework capable of using kernel density estimation on the categorial space. The difference between the Bayesian optimisation in automatic chemical design and Gryffin is that the Bayesian approach has to learn a map between the categorical inputs spaces and continuous ones, while Gryffin is able to directly search the categorial space with no need to learn from a map.

Chimera

Häse et al. [30] created an a solution that is able to use multi-objective optimization for computational and experimental design, named Chimera. This optimization solution is aimed to overcome two constrains while deploying self-driving labs. First constrain refers to the timely and costly evaluations, both experimentally and computationally, while evaluating objectives. The second constrain refers to the lack of prior knowledge about the surface of the objectives.

Gemini

Hickman et al. [31] created Gemini, a flexible and computationally efficient tool based on a regularized neural networks that is able to leverage inexpensive evaluations and does not rely on a sizable dataset containing expensive evaluations. The main goal of Gemini is to be used for regression when multiple sources of data (inexpensive) and the expensive training data is limited.

Golem

Golem is an optimizer developed by Aldeghi et al. [32] that aims to find optimal solutions for input variability. It is focusing on 2 types of input variability, the first one being the control factors that can be caused by the experimental protocols or by imprecise instruments. The second input variability that can alter the performance of the optimization is due to conditions in which the experiments is carried out, such as the humidity or the temperature of the room. Taking the input variability into account, Golem is using a probabilistic approach identifying optimal solutions that are robust to uncertainty.

Table 2.3 provides an overview of some of the most commonly used planning algorithms in the field of material acceleration discovery.

Algorithm	Author	Characteristics
Phoenics	Häse et al. [28]	Continuous variables, Single-objective optimization, BNN based
Gryffin	Häse et al. [29]	Categorical variables, Single-objective optimization, BNN based
Chimera	Häse et al. [30]	Hierarchy based Multi-objective optimization, BNN based
Gemini	Hickman et al. [31]	Scalable multi-fidelity ML
Golem	Aldeghi et al. [32]	Input variability optimization, ML based

Table 2.3: Related work on planning algorithms and their main characteristics.

2.2.4 Application Fields

The discovery of new materials can boost multiple fields of research. Until a few years ago has generally been done manually and based on human intuition, but work is now underway to develop automated applications to discover new materials using ML to design better experiments with the knowledge currently available. Some of the areas where the use of MAP can foster growth include green energies, synthesis of organic and inorganic molecules and nano-structures such as quantum dots among others.

Power-to-X (PtX) encompasses a range of conversion technologies that leverage electricity to produce carbon-neutral fuels including hydrogen, synthetic natural gas and liquid fuels. These synthetic fuels can contribute to decarbonising sectors where this was previously practically unfeasible or can be stored for later use, unlike electricity.

In this field, Langner et al. [33] used high-throughput experimentation (HTE) and a self-driving laboratory to optimize the composition of multicomponent blends for organic photovoltaics (OPVs), resulting in increased efficiency and stability.

Progress has also been made within the PtX field in the improvement of catalysts to enhance the hydrogen production process, an example being the novel approach presented by Bai et al. [34], which combines robotic experimentation and high-throughput computation to explore the potential of conjugated polymers as photocatalysts for hydrogen production. In this reasearch over 170 co-polymers were synthesized and characterized, as shown in the workflow in Figure 2.15, leading to the discovery of new polymers with high sacrificial hydrogen evolution rates.

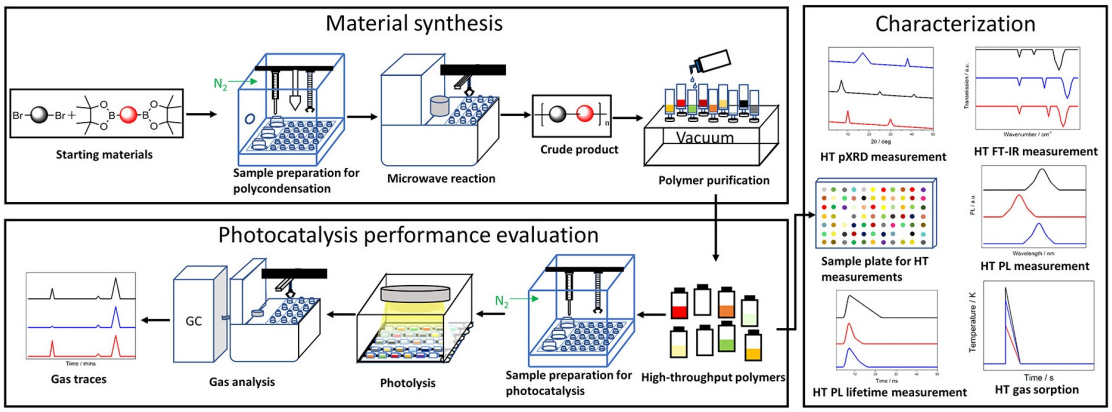

Figure 2.15: Workflow for synthesis and property characterization of the conjugated polymers by Bai et al. Reproduced under a Creative Commons Attribution 4.0 International (CC BY 4.0) from Ref. [34], Copyright 2019

Some applications of MAPs and self-driven laboratories are emerging in the field of synthesis of new organic materials with different applications. Wu et al. [35] developed an automated platform to discover new organic molecules to be used as laser gain mediums. They use an end-to-end approach consisting of 3 parts, organic synthesis, product purification and identification and finally, optical characterisation.

Another achievement is the one presented by MacLeod et al. [12], they developed a platform capable of processing, synthesizing, and characterizing organic thin-films autonomously. The motivation for such a development is that the optimization of thin-film materials is very time consuming due to the large number of existing compositional, deposition, and processing parameters. The platform is autonomously finding new experiments using Phoenics (explained in subsubsection 2.2.3) and actively learning from past data. The platform is able to dispense, aspirate and mix liquid precursors as well as characterizing the films using ultraviolet-visible-near-infrared reflection and transmission spectroscopy. By using this system, MacLeod et al. [12] were able to synthesize and characterize a sample every 20 minutes, opposed to other solutions that can take up to 60 minutes [36].

Although much more effort has been put into the synthesis of organic materials, there is also some work such as Szymanski et al. [9] which focuses on automated inorganic synthesis using solution-based routes, thin-film deposition and solid-state reactions.

Progress has also been accomplished in the area of quantum dots (QD), a type of nanometric crystals used as semiconductors that have unique optical and electronic properties [37]. QDs are used in different areas such as: light-emitting diodes, photovoltaics, and biomedicine. A common usage of QDs is in LED TV displays, called QLED. Since the QD can be synthesized, and there already exists a number of synthesized QD used for different applications [38], there is a motivation to further research and discover new ways to synthesize QD.

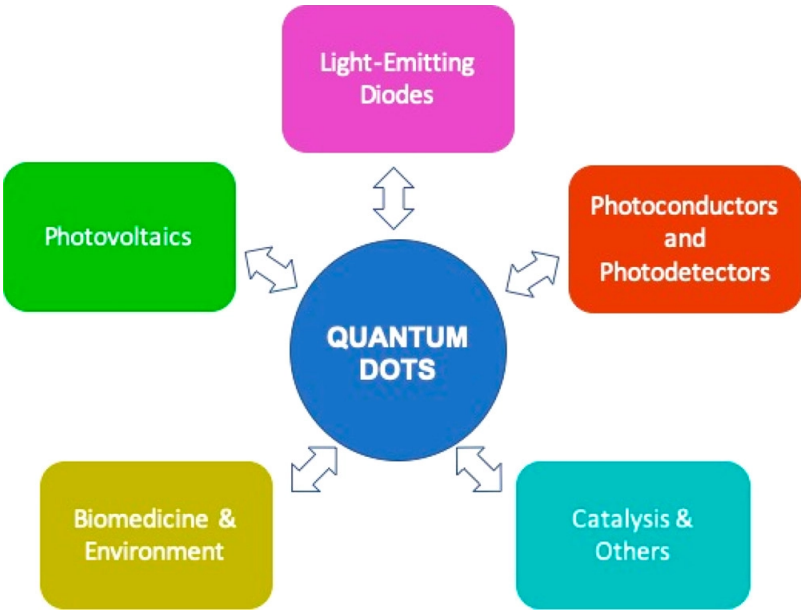

Figure 2.16: Applications of Quantum Dots [37]. Reproduced under a Creative Commons Attribution 4.0 International (CC BY 4.0) from Ref. [37], Copyright 2020

Different teams are working towards the acceleration of the discovery/synthesis of QD. Abdel-Latif et. al. [39] are working on a system that is capable of synthesising inorganic lead halide perovskite QD using AI in hopes of reducing the time it takes at the moment, due to the vast complexity number of parameters. Epss et. al. [10] proposed a new autonomous material discovery system aimed for perovskite quantum dots in hopes to overcome the obstacle that, for 5 years, is slowing the development of such a material, namely the trial and error-based QD synthesis.

Table 2.4 shows the related work in the field of MAPs and its applications in recent years. It can be seen that the possibilities of the new materials discovered are very broad and diverse.

Author	Year	Application
2010	Konstantatos et al. [40]	Nanostructures for photo detection using colloidal QD
2010	Chan et al. [13]	Synthesis of colloidal inorganic nanocrystals
2013	Chen et al. [41]	Photovoltaics nanochemstry using QD
2017	Ramasamy et al. [42]	Color tunable QD LEDs
2020	Langner et al. [33]	Self-driving laboratory for HTE for OPV
2020	Amaral et al. [43]	Biomedical cell Imaging fibrous phosphorus QD
2020	MacLeod et al. [12]	Accelerated discovery of thin-film material
2020	Epss et al. [10]	Accelerated process of discovery for metal halide perovskite QD
2021	Wagner et al. [14]	Controlling multiple MAPs. Main focus on thin-film materials
2021	Szymanski et al. [9]	Automate inorganic synthesis
2023	Wu et al. [35]	Discovery of organic molecules for laser gain mediums

Table 2.4: Related work within the field of MAPs.

2.3 Alternative solutions towards MAPs

Alternative production strategies will be considered in this section for the implementation of the material acceleration platforms.

2.3.1 Swarm production

Swarm robotics is an emerging field of robotics with the potential to transform manufacturing processes. This novel approach involves the coordination of multiple (usually simple) robots as a system. The robots in a swarm robot system usually have limited sensing and communication capabilities. Thus, through local interactions between the individual robots and the environment the collective behaviour of the robots is achieved [44]. By integrating swarm intelligence with traditional robotic techniques, it enables large, complex structures to be manufactured more cost-effectively [45].

This approach is evolving rapidly but for now there are hardly any real applications of robotic swarms for production and manufacturing. At present, their applications are mainly in areas such as navigation, coordination and collective decision-making. Recent developments include the creation of models to predict the progression of a robotic swarm or studies to determine how robots can collectively decide based on individual preferences in the best-of-n problem. [46]

There is still a long way to go in this promising field. It is therefore to be expected that the coming years will see advances in swarm robotics for manufacturing inspired into natural swarming behaviour. [47]

2.3.2 Flexible Manufacturing systems

Using a Flexible Manufacturing System (FMS) allows for the rapid alteration of a product in an autonomous manner. The FMSs are heavily reliant on Industry 4.0 technologies such as the Internet of Things (IoT) where the machines present on a production line are able to communicate with each other. The main purpose of a FMS is to enable a production line to deal with high product variation and/or short product cycles while being able to provide cost-effective, high quality and consistent results. [48]

Since creating new production lines can be a considerable investment for a company, FMS has to be flexible enough to handle small batches based on customer demand. Because of this flexibility, human workers have the job to maintain these systems by changing tools, loading and unloading different parts, based on the requirements of the next batch. [48]

Due to the usage of IoT, FMS can incorporate digital twins that can be used in online simulations of machines. Using digital twins in a production line can bring several benefits for the company, such as foreseeing potential issues or predicting different metrics during production. Having a continuous data flow between the physical equipment and

the digital twin, human workers can get an overview of the status of the production and take decisions accordingly, improving the response time in case of failure or other issues that may arise, which would not be noticed until it is too late if the digital twin and FMS would not be used. [48] Moreover, a digital twin can be used in the design stage of a production line which can help visualise a product in the early stages when different aspects of the production have to be established. [49]

Looking at Figure 2.17, different manufacturing approaches provide different levels of versatility and performance. Having a highly flexible production line can have a reduced productivity rate since the production line is build to handle different types of products, while a production line that focuses only on a product will have a high productivity, but it will lack any flexibility.

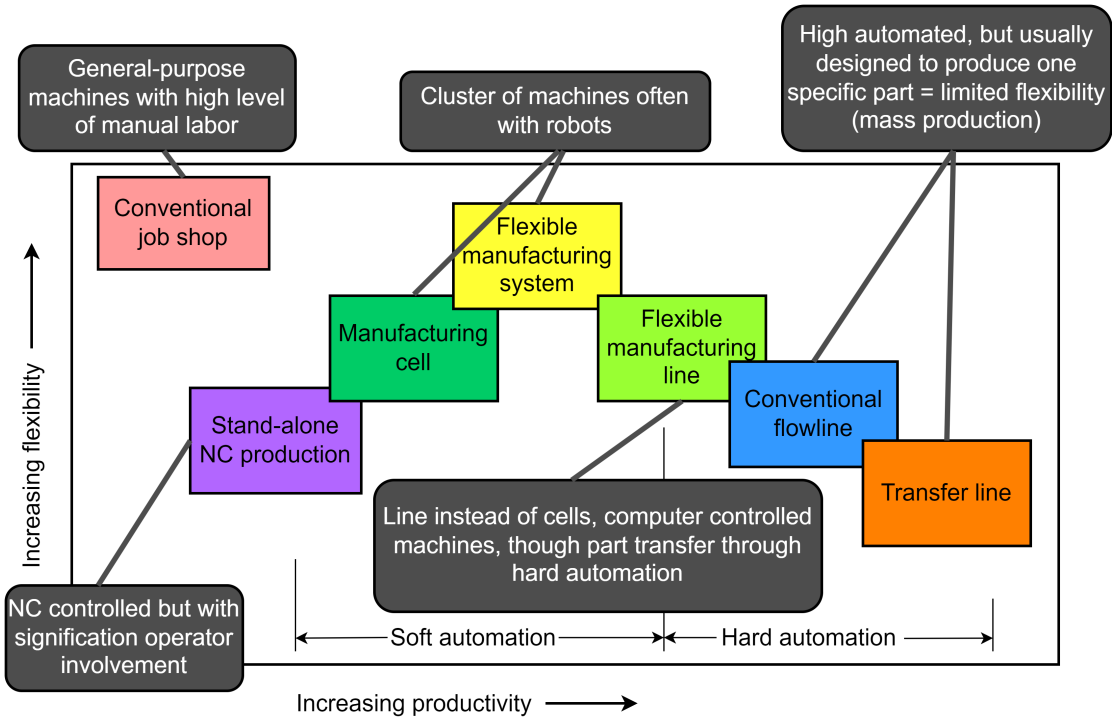

Figure 2.17: Flexibility and productivity of different manufacturing approaches. Adapted from [50].

2.4 Actual Use Cases in the Laboratory

A visit is made to the laboratory to observe the processes performed by chemistry students. Their aim was to measure which steps are the most common to introduce errors in simple laboratory processes. For this purpose, 2 different dissolution processes are carried out, using demineralised water and acetylsalicylic acid as agents.

In the first process they conduct "Standard Series", which is considered to be the more precise of the 2 methods. For this experiment, dilutions with 4 different concentrations are made, and 2 samples are collected from each dilution. The detailed procedure is as follows:

- Calibration of flasks and pipettes (In most experiments in the laboratory this calibration is not carried out), Figure 2.18 A.
- Weighing the compound (acetylsalicylic acid) on a special sealed balance so that the negative pressure of the environment does not affect the measurement, Figure 2.18 B. The precision of the digital scale is 0.1 mg, therefore, in this step the uncertainty in the measurement is added to the process, which is equal to the resolution.
- Pour the compound in corresponding flask.
- Water is added, at this step there is a possibility of adding a parallax error to the process.
- Dissolve by magnetic stirring (visual inspection for readiness), Figure 2.18 C.
- Transfer the solution from the flask into a beaker.
- Filter the solution with a syringe into the HPLC vial (it is not necessary to be precise in the amount of solution added in this step), Figure 2.18 D.
- Sealing of HPLC vials with a dedicated tool.
- Placement of vial containers.
- Adjust the settings of the HPLC program.
- This process is carried out 5 times in total.

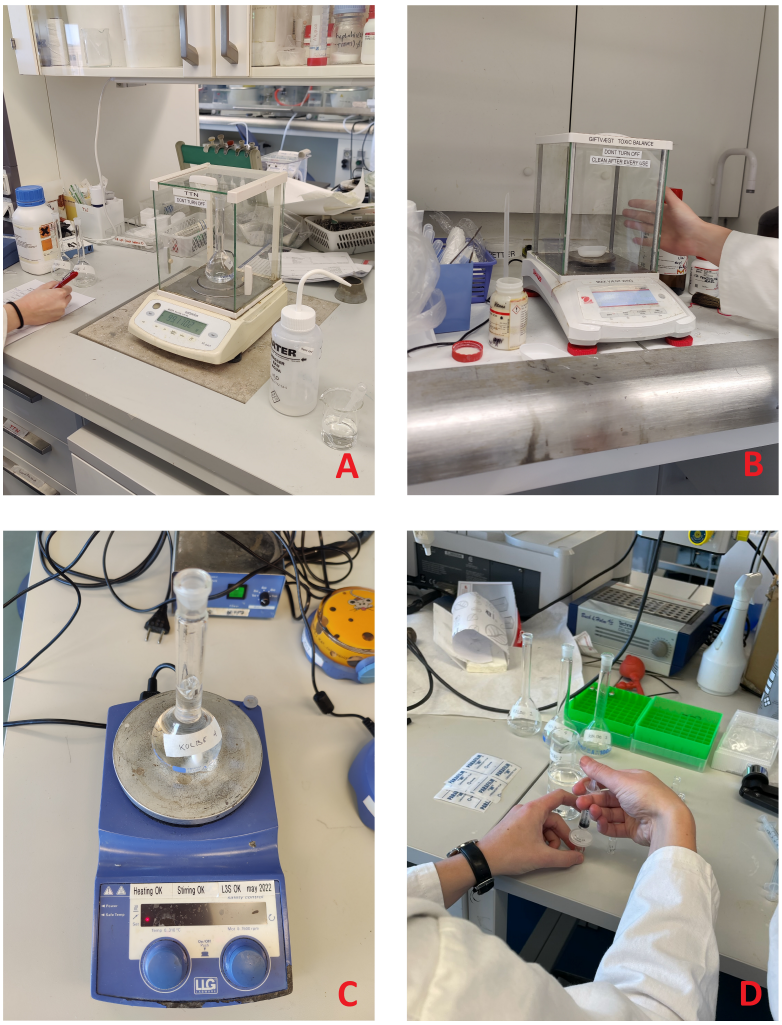

Figure 2.18: Different steps carried out during standard series experiment. A) Calibration of flasks and pipettes. B) Weighing of acetylsalicylic acid in sealed scale using disposable pans. C) Compound mixing by magnetic vortex stirrer. D) Filtering of the final solution with a syringe into the HPLC vial.

The second experiment, or "Dilution Series", is commonly used in the industry although it is considered slightly less accurate. The process is fundamentally the same, but instead of adding different amounts of solute, a concentrated dilution is made and different amounts of water are added several times to achieve the desired concentrations. This process is also repeated 5 times, resulting in a total of 140 samples. In this experiment, a pippeting step is added to add the exact amount of demineralised water (Figure 2.19), which is a tedious and time consuming process.

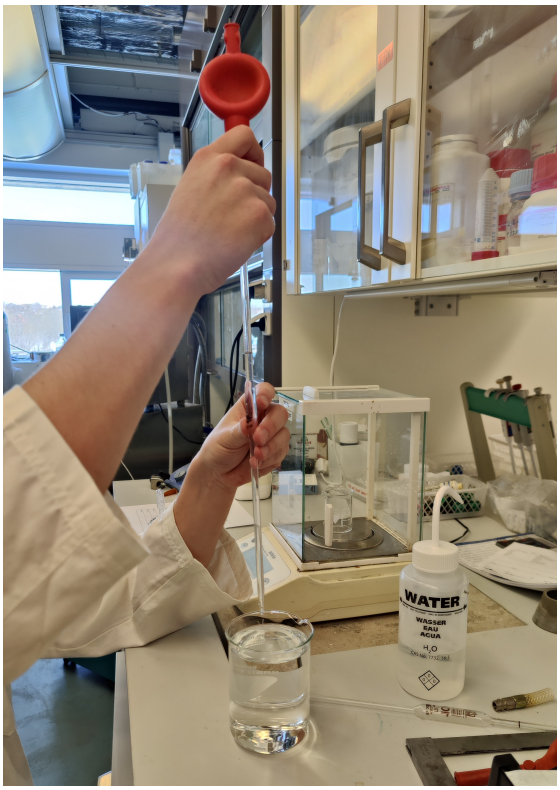

Figure 2.19: Extraction of exactly 5 ml of water with a precision pipette. It was observed that the collection process took on average more than 30 seconds.

The conclusions drawn from the laboratory visit can be summarised as outlined below:

- Considering only the preparation and conduct phases, it takes more than 8 hours to complete the experiment.
- HPLC takes about 30 minutes to get results per sample.
- HPLC analysis of the vials will take several weeks in analysing the 180 samples as this machine is shared in the laboratory.
- Every step is done manually, there is hardly any possibility to automate the processes as performed by humans, they are very complex. A radical redesign of the whole procedure is needed.
- In an ordinary experiment there would be 5-10 times fewer steps because the experiment is only executed 1-2 times, not 5 times.
- In these processes, human errors occur with a high probability.
- During the course of the experiment (excluding HPLC characterization), no specific process has been found to constitute a bottleneck. Each step in the process has great potential for efficiency improvement through automation.
- In certain processes, such as measuring the solute or adjusting the amount of water to be added, it is very complicated to be precise by hand. The smaller the quantity, the more difficult it is to be accurate.

2.5 Simulation versus Real World Environment

While the results of the chemical processes can only be observed in the real world, the development of a self-driven laboratory can still be done with the help of simulation software. In such a software, different aspects of the process can be thoroughly tested, such as the robot movements or the orchestration abilities. By choosing to develop a robotic solution in a simulation software, numerous advantages become available, such as:

1. **Cost.** Compared to the price of setting up a physical system, having a simulation can be cheaper.
2. **Time.** It is less time consuming to build a simulated environment. Moreover, it takes less time and effort to make design changes in a simulation.
3. **Risk.** A simulation is able to provide a safe testing environment for both humans and equipment.
4. **Digital twin/Digital shadow.** Enables the integration of the digital twin or digital shadow concepts.
5. **Development and ideas.** A simulation can provide a sandbox in which different ideas can be safely developed and deployed.

A big challenge that can be considered a disadvantage for using simulation software is the transfer from simulation to reality, which in some cases can be complicated. Some software, mainly off-line programming (OLP) software, incorporates connection solutions for different robotic brands, but others do not offer this functionality and a workaround has to be found.

Both digital twin and digital shadow concepts are highly based on simulations, but the main difference between the two concepts is that a digital twin is able to both receive and send data between the physical world and the simulation, while the digital shadow is able to only receive data from the physical world to the simulation.This is illustrated in the Figure 2.20. [51]

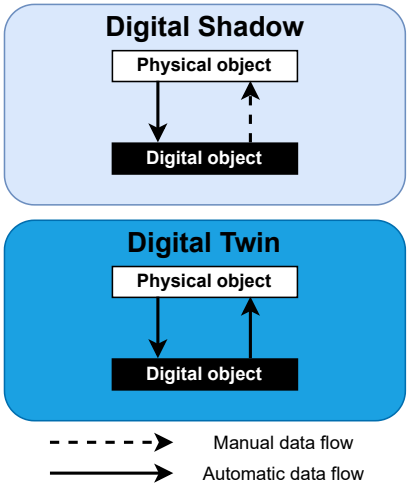

Figure 2.20: Representation of the communication between the physical and the digital world in both Digital Shadow and Digital Twin concepts. Adapted from [51].

For robotic applications there are multiple simulation software available on the market, such us Unity 3D, Gazebo, Webots and Nvidia Isaac Sim. Unity 3D is mostly a game and physics engine mainly used in the video game industry, but due to its capabilities, multiple plugins were made by different open-source communities to support robotics. Gazebo and Webots are mainly used within the robotics discipline, with both of them being available as open-source [52] [53].

Chapter 3

Problem Formulation

This chapter presents the final problem formulation, an explanation of the project objectives and a statement of the requirements for the proposed solution. The solution provided to this challenge will be presented in chapter 4.

3.1 Final Problem Formulation

How can the chemistry laboratories be redesigned by using simulation, robotic manipulators and magnetic levitation platforms with the purpose of drastically accelerating the discovery of new materials?

3.2 Project Objectives

In the upcoming section, an overview of the goals of this project is presented. The objectives are further subdivided into specific tasks that are necessary to address in order to solve the problem posed in section 1.2.

O1: Define the requirements of the project
Two sets of requirements will be defined. The first set is related to the design of the system, where the platform must have diverse capabilities, providing the system with modularity to allow adaptation to the variety of tasks necessary for the acceleration of material discovery. In this way, tools for the most frequent processes must be in the configuration on a near-permanent basis, while tools for more specialised processes can be added as needed. This set of feature requirements will subsequently be used for testing. The second set will focus on the overall solution, which will require defining the characteristics of the system, with priority given to making the project scalable in the future. Some requirements of the solution will be hard set by the nature of the project.

O2: Create a simulation of the AAU Matrix Production system
Considering that our work will be built upon for further progress, the simulation environment must be created using Nvidia Isaac Sim. The simulation environment should be fully functional and should mimic the real setup as closely as possible.

O3: Optimise the behaviour of the physical setup using AI-Orchestration
There is software that uses AI algorithms to orchestrate experiments to discover new materials, the aim is to adapt the operation of this algorithms to control the robotic arms and the magnetic platform, optimising the efficiency of experiment performance.

O4: Implementing and testing on the physical system
Once the system is ready for deployment, a proof of concept will be run on the physical system. These experiments will also serve as a testing method for the system, where the design requirements can be assessed.

05: Cooperate with research partners and specialists in the fields of chemistry synthesis and/or PtX.
Our solution must align with what the subject matter experts need and solutions should be sought to the bottlenecks they often face. The way in which experts conduct experiments should not be replicated, however, ways should be found to carry them out as efficiently as possible using the technological resources available.

3.3 Requirements

The subsequent section will explain in detail the two sets of requirements established for the project based on the objectives set out in section 3.2. Prior to addressing the requirements of the project, it should be noted that this project is a proof of concept and no real chemical experiments will be carried out.

3.3.1 Design Requirements

Based on the findings presented in chapter 2 and based on the use case presented in section 2.4 the robotic platform must have the following capabilities to be able to recreate the presented experiment:

- Dispense solids
- Dispense liquids
- Aspirate
- Stirring
- Pick and place
- Transportation and handling

Moreover, adding the following requirements will allow for the scalability of the platform for a wider range of experiments:

- Annealing
- Heating
- Separation
- Decomposition

3.3.2 Solution Requirements

The following requirements are used to evaluate the solution presented in chapter 5

S1: The solution must be able to recreate the real setup in simulation
Since this project will be a stepping stone for future research at AAU Matrix Production, Nvidia Isaac Sim is a simulation software that is wanted to be implemented and made standard. The physical setup in the lab must be recreated in this simulation software.

S2: The solution must be modular
The MAP system must be able to conduct a wide variety of chemical experiments, it is therefore necessary to provide it with a certain degree of modularity.

S3: The solution must be able to perform an experiment in the real setup
The implementation must include the control of the physical devices.

S4: The solution must be able to carry out an experiment given certain instructions
The solution must be able to execute machine readable instructions.

S5: The solution must integrate AI-orchestration for the planning and execution of experiments
The solution must be able to work in conjunction with existing chemical experiment orchestrators.

Chapter 4

Implementation

This chapter presents the implementation of the proposed solution and is divided into multiple sections. The Design section provides an overview of the implemented configuration and explores alternative options. The Simulation section focuses on the experiment environment in Isaac Sim. The Control section explains the applied methods for controlling the shuttles, robotic arms, and experiment orchestration. Lastly, the physical setup implementation is thoroughly explained.

4.1 Design

Figure 4.1 depicts the workflow diagram showing the sequential steps involved in the chemical synthesis process presented in section 2.4. The main design of this project will be based on the AAU Matrix Production setup shown in Figure 4.2.

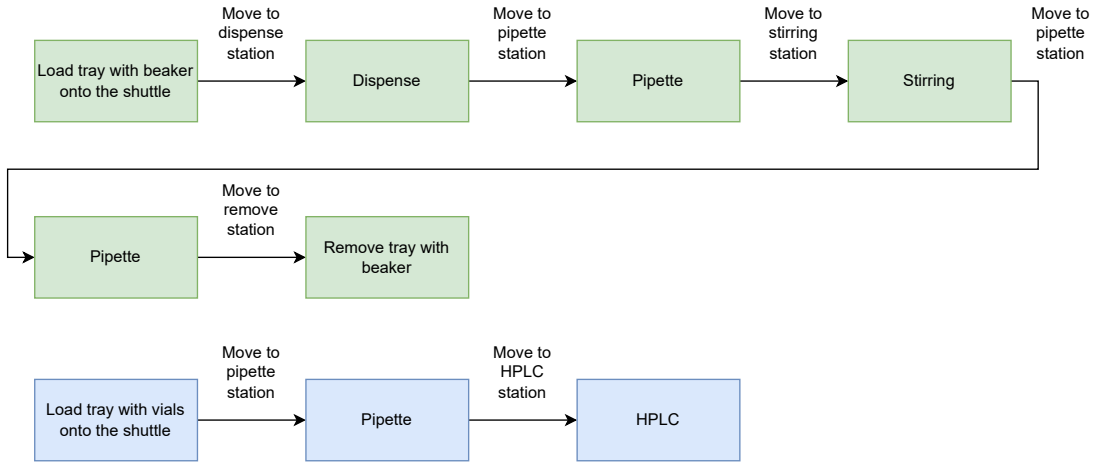

Figure 4.1: Experiment workflow design. The experiment will be based on these two concurrent workflows. The green workflow represents the process associated to the beaker, while the blue workflow represents the process related to the vials.

Figure 4.2: AAU Matrix Production using the Acopos 6D platform. Instead of using 6 robots as shown in , the implementation will be done on 5 robots instead. This is because the 6th robot (top right in the figure) was added towards the end of the project and including it in the implementation was not feasible anymore.

Before creating a design, the robot's reach and the measurements of the magnetic segments have to be considered. The Kuka KR3 R540 has a reach of 541mm [54] from the middle of the base and the Kuka KR4 R600 has a reach of 601mm [55] from the middle of the base as well, and the magnetic squares have a measurement of 120x120mm.

The workflow depicted in Figure 4.1 is explained in detail in Figure 4.4, Figure 4.5 , Figure 4.6 and Figure 4.7; and Figure 4.3 shows the legend for each component. The workflow starts by placing the tray containing a beaker on the magnetic shuttle and then moves to the dispensing station a solid compound is being dispensed into the beaker.

Once the beaker is filled to the required quantity, the shuttle goes to the pipette station where the solution is added. When this step is finished, the solid compound and the solution are ready to be mixed, hence the shuttle moves to the next station, where a manipulator will place the tray containing the beaker on a stirring machine.

When the stirring process is done and the compound is dissolved, the tray is placed back on the shuttle and it goes back to the pipette station, where the solution is aspirated and it will later be poured into the vials. Once this process is done, the shuttle containing the empty beaker moves to the removing station where a robot will empty the shuttle by picking the tray. At the same time, a tray containing 16 vials is loaded on a shuttle.

While the empty shuttle that had the beaker returns to the initial position, the shuttle with all the vials moves to the pipetting station, where each vial is filled with a liquid solution. When all the vials are filled, the shuttle moves to the HPLC station, where a manipulator will remove the tray and the vials will be ready to be placed in the HPLC machine.

The steps of the solution workflow are represented in the following figures, with the aim of present a visual representation of the process.

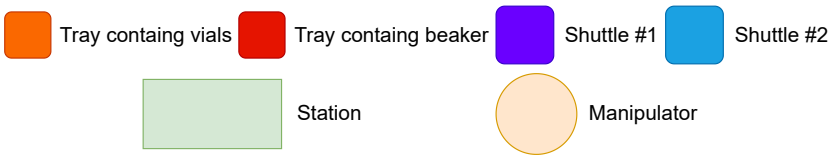

Figure 4.3: Component legend for Figure 4.4 to Figure 4.7.

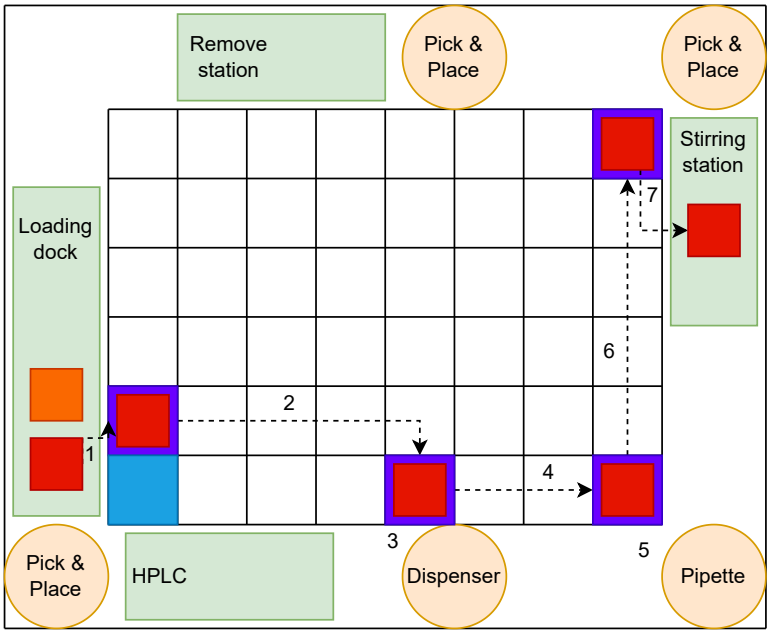

Figure 4.4: First part of the experiment. Shuttle #1 is moving the tray with the beaker from the loading dock to the dispensing station, from the dispensing station to pipette station, from pipette station to stirring station.

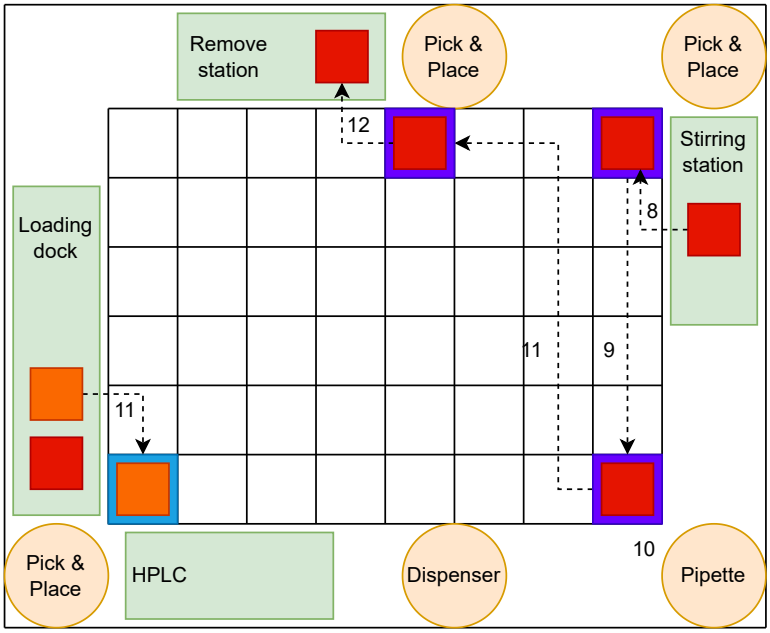

Figure 4.5: Second part of the experiment. Shuttle #1 continues to move the tray containing the beaker from stirring station back to the pipette station and finally, from the pipette station to the remove station, where the tray is removed from the shuttle #1. While the shuttle #1 is at the pipette station, shuttle #2 picks up the tray containing the vials.

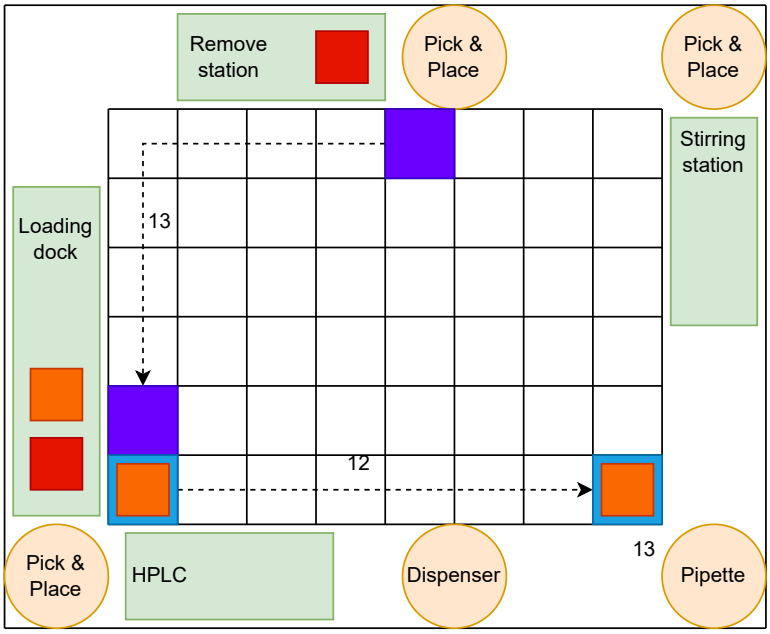

Figure 4.6: Third part of the experiment. Shuttle #1 return to the initial position while shuttle #2 moves to the pipette station.

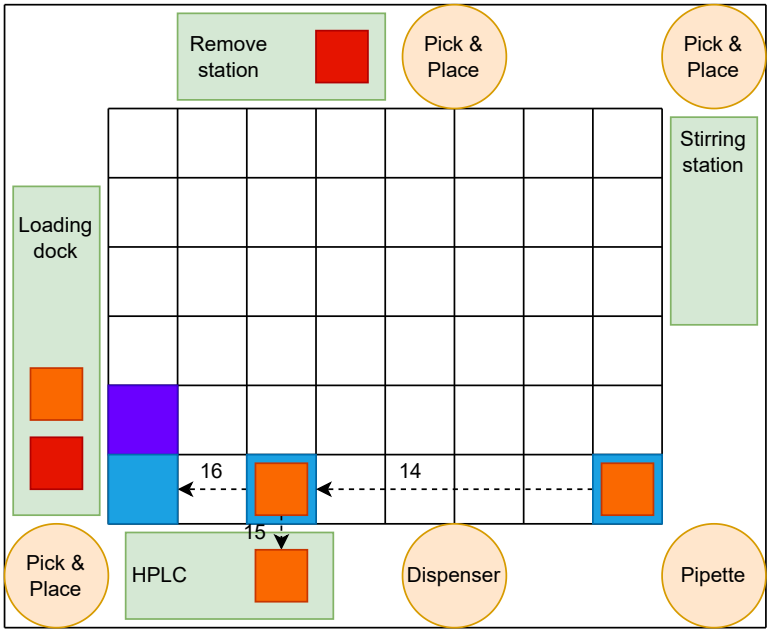

Figure 4.7: Forth part of the experiment. Shuttle #2 moves the tray with the vials from the pipette station to the final station, the HPLC station.

Analysing the workflow presented from Figure 4.4 to Figure 4.7, it is observed that the "Pick & Place" robots for the "Remove station" and for the "Stirring station" have only one task each, and one of the robots could take the responsibilities of the other.

By implementing this change, it would enable one of the robots to execute a parallel process, resulting an enhanced system efficiency or remove one robot and part of the platform, resulting in a reduced cost. Moreover, the fact that the task can be accomplished even after removing one robot demonstrates the inherent modularity of the system.

4.1.1 Alternative designs

Simplified version of the main design

An alternative design is considered, where the same task presented from Figure 4.4 to Figure 4.7 can be achieved using 4 manipulators instead of 5. Additionally, the size of the magnetic platform is reduced.

In this design, the robot responsible for removing the trays from the shuttle has been removed. Now, the task of removing the trays is assigned to the manipulator located next to the stirring station. This simplified version of the main design is depicted in Figure 4.8 and Figure 4.9.

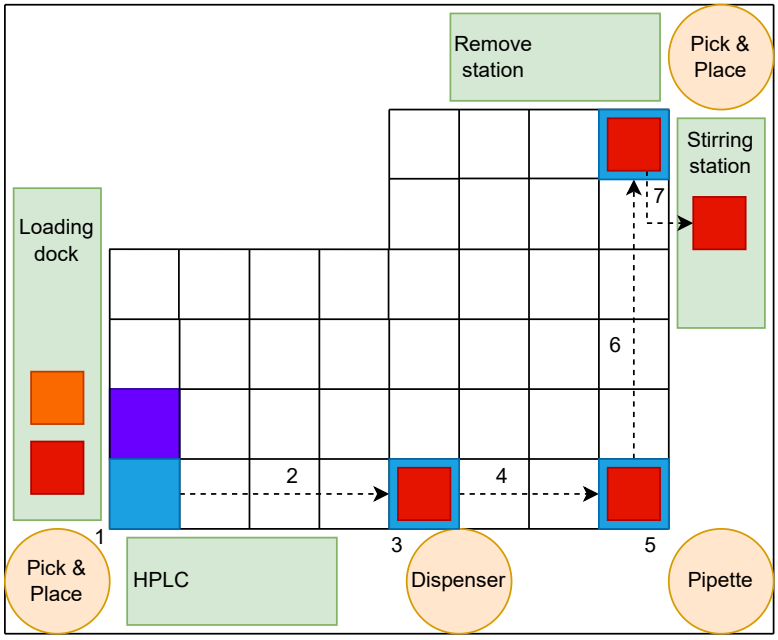

Figure 4.8: First part of the alternative design using 4 Kuka manipulators and a smaller magnetic platform.

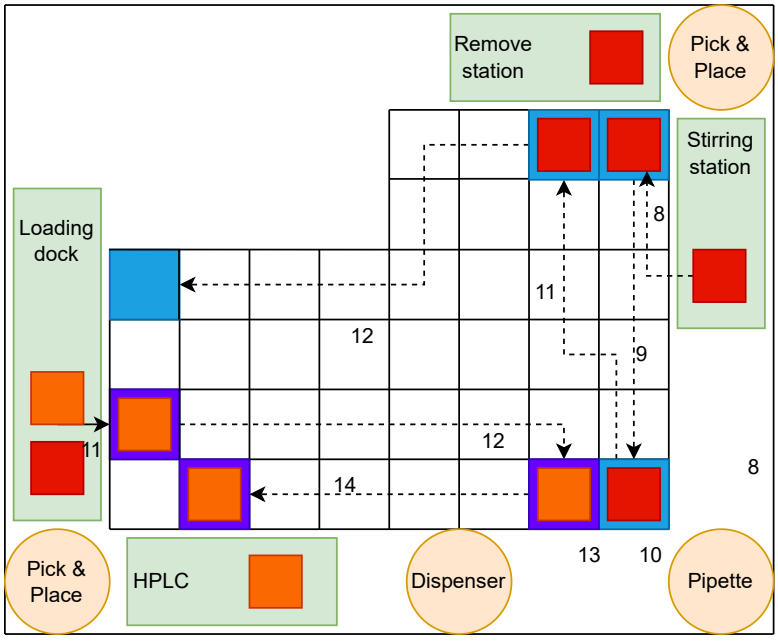

Figure 4.9: Second part of the alternative design using 4 Kuka manipulators and a smaller magnetic platform.

Modular design

Although it is possible to remove and move around parts of the platform as shown in subsubsection 4.1.1, another alternative design is considered.

In this proposed design, the main focus is on modularity, where multiple MAP platforms can be connected to form different shapes, as depicted in Figure 4.11 and Figure 4.12. The design of this system resembles to the configuration of Festo modules in the AAU Smart Production Lab, as depicted in Figure 4.10.

This approach could be better tailored for different laboratory layouts or for different chemistry processes. Using a mobile robot, as in Figure 4.12, could increase the level of automation by delivering the samples to different parts of the laboratory.

Additionally, by placing the MAP system in a clean room, the presence of humans in the room can be reduced with the assistance of a mobile robot. Working in clean rooms entails strict regulations for individuals, including the requirement of slow movements to prevent the emission of human particles [56] and limited time spent inside the room. Consequently, transferring work load to a mobile robot can be very beneficial.

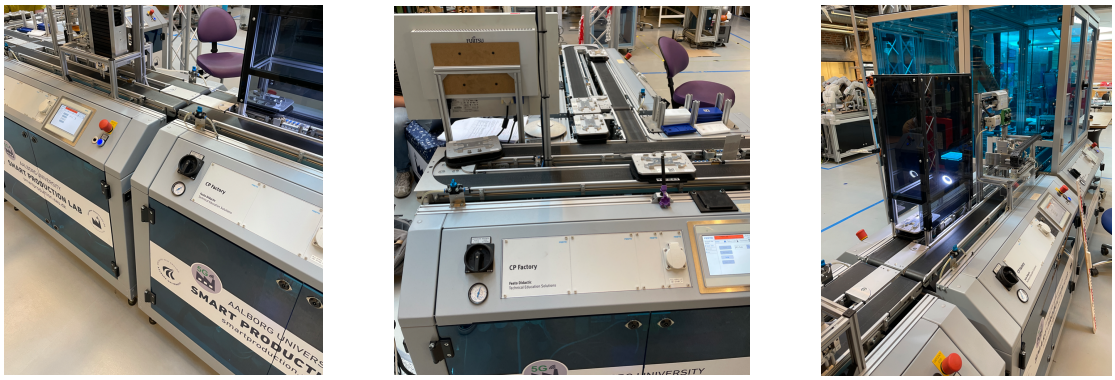

Figure 4.10: Different Festo modules that can be connected together into different shapes.

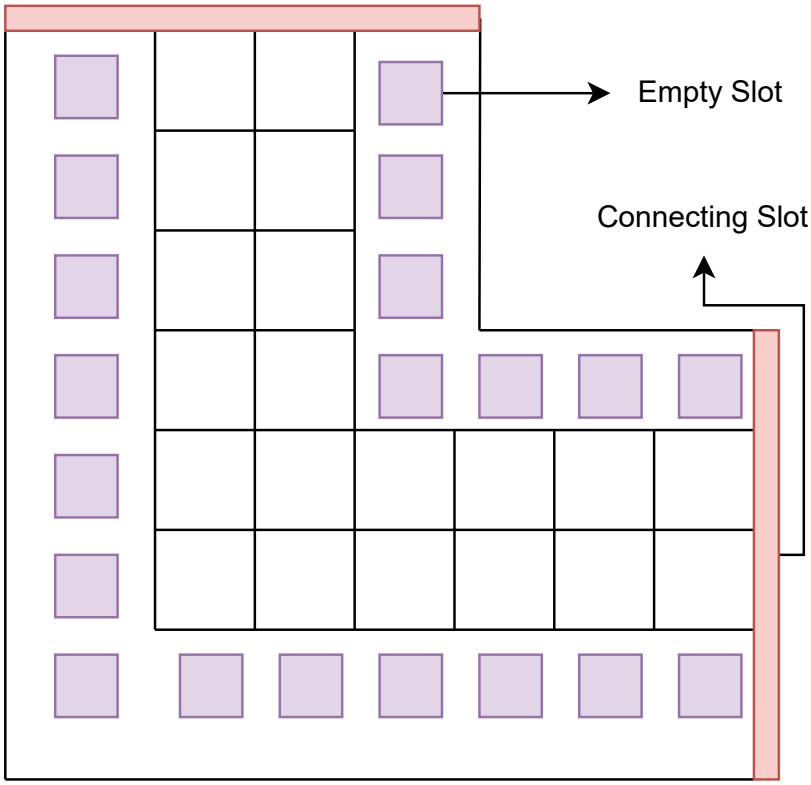

Figure 4.11: Modular approach where robots and different stations and/or utilities can be mounted to the empty spaces (purple slots). The platform can be extended by connecting another platform to the connection places (light red slots).

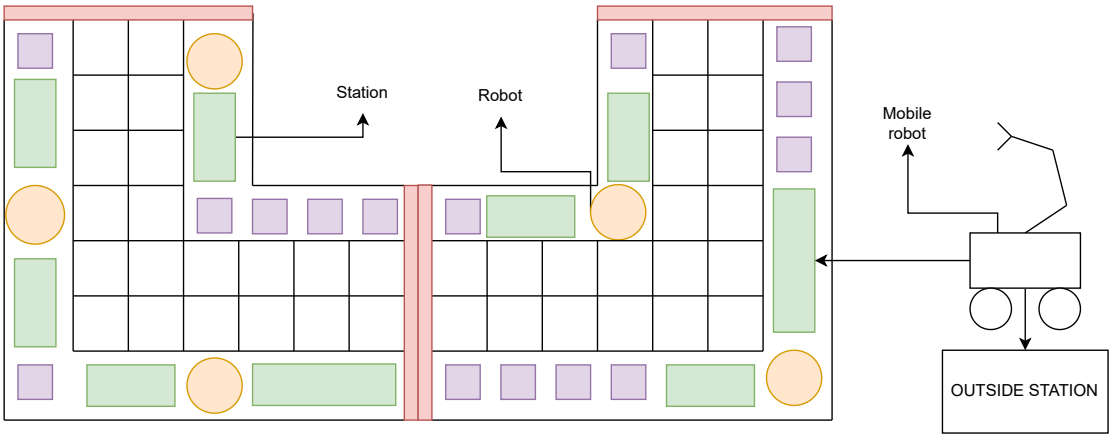

Figure 4.12: Example of an arbitrary MAP setup consisting of 2 platforms, each equipped with different stations and robots. A mobile robot can be used to connect the MAP setup to any outside station where different process can be carried out.

4.2 Simulation

Due to the requirements of the project, Nvidia Isaac Sim is the simulation environment of choice and no other simulation software was considered. Furthermore, in this project, Robot Operating System (ROS) is used to close the gap between the real world and the simulation.

4.2.1 Nvidia Isaac Sim

Isaac Sim is a simulation and data generation application for robotics able to create photorealistic and physically accurate virtual environments [57]. Its features include scalability and ease of AI integration, as well as Python programming and ROS compatibility.

Isaac Sim is integrated into Nvidia's Omniverse platform [58], which offers numerous applications for 3D workflows and applications built with USD (Universal Scene Description). USD is an open source file format for describing 3D scenes developed by Pixar [59]. It allows to work collaboratively on different aspects of a scene simultaneously. USD is efficient, scalable and useful for animated films, visual effects and real-time simulation projects.

4.2.2 Isaac Sim Extensions

Isaac Sim provides several extensions that simplify the creation of a virtual environment by streamlining the implementation of robotics in a simulation.

URDF importer

This extension allows the import of Universal Robot Description Format (URDF) files for conversion to USD. The URDF file contains different parameters for robotic systems, such as 3D models, collision models, dynamic parameters, joint limits, masses and other meta-data.
This tool is used to import different robotic arms to obtain their equivalent in USD format.

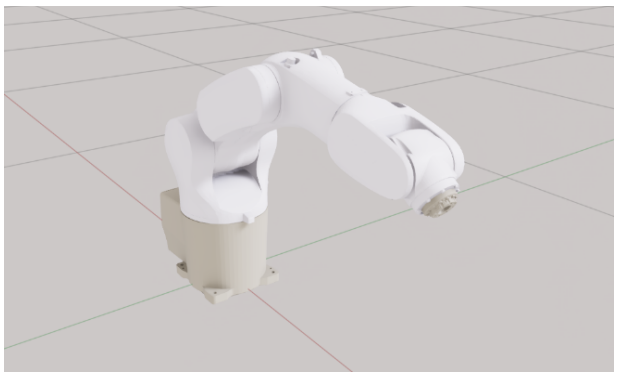

Figure 4.13: Robot Arm Kuka KR4 R600 imported from URDF file to USD in Isaac Environment.

Gain Tuner

The gain tuner utility extension in Nvidia Isaac Sim allows for stiffness and dumping tuning on the fly. For the extension to work, a robot manipulator or any other system that has at least one joint has to be loaded in the world and the simulation has to be running.

Once these conditions are met, in the gain tuner extension, the system that has the joints will be available for selection and then a window showing all the joint values appears. In here, different values can be chosen for all the joints at the same time by using the scaling function, or by manually inputting a value for each joint.

For testing, there is a function that will send random joint values for the manipulator and it can be visualised how the robot will perform with the set values. Once the movement is suitable, the USD file can be saved with the set parameters.

Articulation Inspector

Similarly to the gain tuner, the articulation inspector, works on the same principle, a system having at least one joint and a running simulation are required. This tool is useful for validation purposes, since it receives all the parameters of a joint meaning that the presence of a joint can be tested as well as the joint limits or other different joint specifications.

Action Graphs

Nvidia Isaac Sim is using action graphs to define and plan the behaviour of a robotic system within the simulated environment. It works by creating a sequence of different actions that are activated based on the output of the antecedent action. In this project, the action graphs, Figure 4.14, are exclusively used for the ROS bridge between Isaac Sim and Moveit.

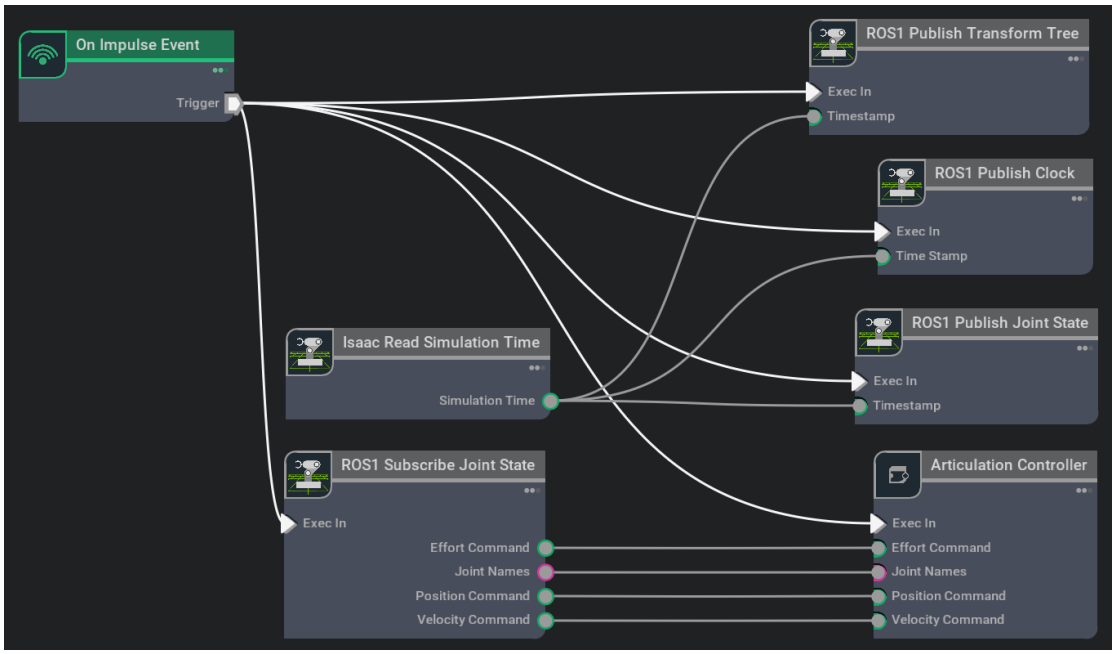

Figure 4.14: Action graph for ROS communication. The articulation controller is getting the joint information (effort, name, position and velocity) from the ROS1 Subscriber Join State block that is connected to the /joint_command topic. At the same time, Isaac Sim is publishing to the /joint_states node so Moveit is able to know the position of the manipulator in the Isaac Sim.

4.2.3 MAPs Simulation Environment

There are several ways to create a simulation environment in Isaac Sim, it is possible to work using the GUI in an intuitive way without programming, it is also possible to create the simulation environment as an extension that runs asynchronously with an interactive interface through python programming, or it is possible to program the environment as a standalone python script with full control over physics and rendering time that can be run headless. This project is built as an extension (MAPs extension) because it provides an intuitive way of interacting with the environment with the advantage of having full control over the simulation.

The extension is responsible for loading all the assets into the simulation in their corresponding position, orientation and scale. The assets are USD files that are located in a designated local path (Omniverse Nucleus could be used to have all the required files in a server). The simulation environment is shown in the Figure 4.15

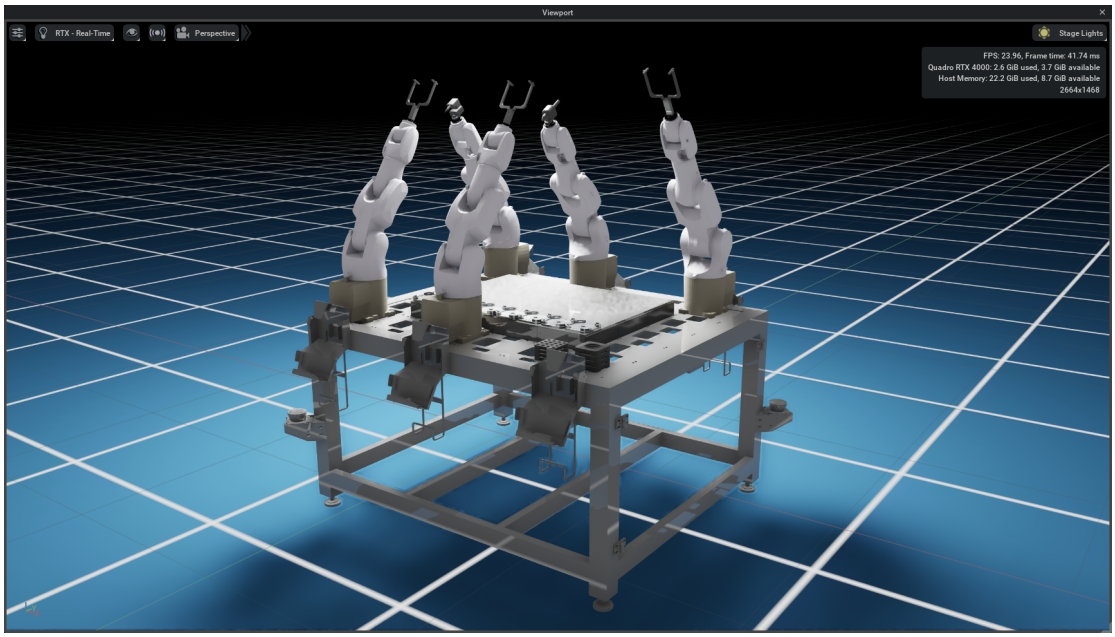

Figure 4.15: AAU Matrix Production laboratory setup in Isaac Sim.

In addition to the assets that make up the setup, other elements must be initialised, such as the action graph that will control the robotic arms. Isaac Sim has its own workspace in ROS and allows interaction with external nodes using the ROS Bridge extension (available for both ROS1 and ROS2). The action graph created works under *OmniIsaacRosBridge* node, subscribing the messages coming from */joint_command* topic and publishing the current position of the robotic arms in */joint_states*.

In the initial development, the intention was to have the control of each robotic arm independently within a namespace, using its own MoveIt config and its corresponding action graph. The idea had to be discarded due to MoveIt's inherent limitations. The solution is to control all the robotic arms as if they were a single robot, this is explained in more detail in subsubsection 4.3.1.

Once the simulation environment has been set up with all the necessary elements, physics callbacks need to be invoked in order to run the simulation. By default a δt of $\frac{1}{60}$ is used for the physics step, i.e. 60 Hz. For the default rendering δt is 60 Hz as well. These callbacks activate or deactivate the various functions that are invoked at each physical step, such as the action graph for the robotic arms, the movement of the shuttles in the simulation or the control and status of the platforms in the real setup. Working in an asynchronous environment, these callbacks can be added or removed using the graphical interface when the simulation is running.

The graphical user interface (GUI) allows interaction with the simulation. For this application it is possible to choose whether to control only the simulation or also the real setup. It allows to connect the simulation to the PMC (subsection 4.3.2) and start the experiment when desired.

In order to make the simulation work as desired, numerous functionalities have been created that allow the experiment to be tweaked in a simple way. To create the simulation environment as well as all the required components to run the simulation, *setup_scene* and *setup_load* are used respectively. These are Isaac Sim's default functions that are executed when the extension is launched. Several functions are then implemented to perform the experiment tasks, such as moving the robotic arms to the desired pose, moving the desired tray to the target position or opening and closing the gripper among many others. These are invoked from the GUI. For more details on the functions implemented and usage, see section A.1

4.2.4 Object interaction in the simulated environment

A significant part of the project is related to the ability of the manipulators to have the pick and place functionality. To achieve this functionality in the Nvidia Isaac Sim, the manipulators (including the gripper), the table, the trays and the shuttles are required to have a rigid body element as well as a collision element. The rigid body element allows objects to be affected by physics, such as gravity. In this project, this allows the trays to be placed on the shuttles without passing through each other.

In Isaac Sim, as in most realistic simulators, there are 3 sets of parameters that affect the behaviour of the simulation. These are the collision geometry, the collision parameters and the physical properties of the material.

Collision Geometry

The collision geometry between objects in the environment is defined by the object's mesh. There are many types of meshes available in Nvidia Isaac Sim that can be used for different applications. There are simple meshes, such as bounding cube (or sphere) which creates a cube (or sphere) around the object, disregarding any appearance details.

The bounding cube can be used for objects that do not require a high level of detail or for the simplification of the environment, since having a complex mesh can be computationally heavy, impacting the performance of the simulation. Another type of collision mesh is the convex hull, which encloses all the outside points of an object by creating straight lines between any pair of outside points. This approach, compared to the simple bounding box, gives a better resolution for the mesh, but slightly decreases the performance. Another similar approach is the convex decomposition that approximates complex mesh shapes by dividing them into multiple convex shapes. This method is particularly useful for representing objects with hollow interiors. A type of mesh that has a high resolution, but at the same time greatly affects the performance of the simulation is the Signed Distance Field (SDF) mesh option. The SDF mesh is creating multiple sets of triangles that are connected through the common edges and vertices. This type of mesh can be helpful in situations where high precision is re-

quired, for instance, when one object has to be placed into another object. Figure 4.16 showcases the difference between 3 types of meshes.

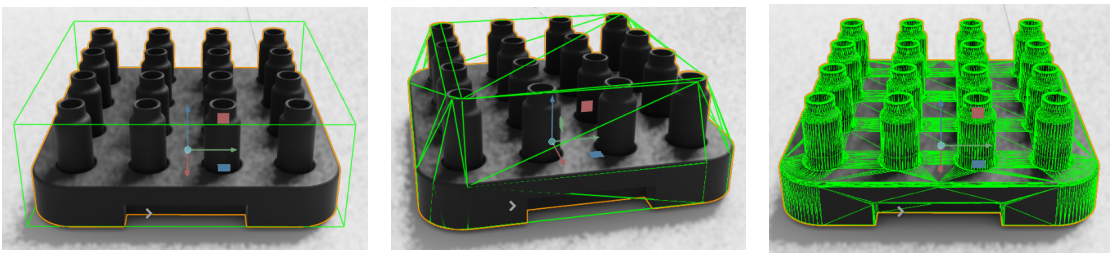

Figure 4.16: Collision mesh examples. From left to right, bounding box, convex hull and SDF mesh.

In this project, bounding boxes are used for the table and for the acopos 6D platform, because the resolution is not important and their only purpose is to hold the other objects. The convex hull is used by the Kuka manipulators, because it gives an acceptable collision resolution that is not computationally heavy. A convex decomposition mesh was chosen for the grippers, which fits the shape accurately enough without being particularly computationally expensive. Finally, a reduced bounding box is used for the shuttle and by default for the tray. Both objects were tested using SDF meshes (as show in Figure 4.17), but the complex geometries created thousands of contact points, which slowed down the simulation enormously.

In addition, the shuttles are moved by each physical step, which at 60Hz caused unwanted jerky movements in the tray even at low displacement speeds. Although this could be solved by increasing the simulation frequency, it still requires an unnecessary increase in computation for this case. For these reasons, a more efficient function has been chosen, according to which, when the tray is inside the shuttle, the movement of the trays will be tied by assigning the xy position of the shuttle plus the corresponding offset at z.

Figure 4.17: Tray containing 16 vials placed on the shuttle using SDF meshes on both objects.

Collision Parameters

Collision parameters in Isaac Sim allow for precise collision control. Depending on the chosen mesh, different collision parameters can be chosen, the most important of which are applicable for all available meshes are collision offset and rest offset. The Collision Offset determines the starting distance for contact generation, while the Rest Offset defines the effective contact distance. Adjusting the Collision Offset can improve collision detection, but excessive values impact performance. Negative Rest Offsets ensure accurate visual alignment when the visualization mesh is smaller than the collision geometry. For our application, negative values are assigned to both parameters.

Material Properties

In order to be able to perform the grip in the simulation, two main properties must be considered. The first property is the mass or density, and must be greater than 0. The density of the material is necessary for the object to behave as a rigid solid. This can be calculated automatically by considering the mass and volume of the material. For this reason, without a defined mass value, Isaac Sim will default to 0 which will make grasping impossible as the physics will not work without a density. It is also possible to assign a value to the density of the material directly. The mass for the gripper fingers was set to 0.15Kg and the mass for the trays was set to 0.35Kg.

The second property is the coefficient of friction of the material, in this case the coefficient of friction of PLA is particularly relevant as it is the material used for gripping in the simulation. Since the friction coefficient of PLA can vary [60], an approximate value of 0.42 was chosen and implemented in the trays and grippers in Nvidia Isaac Sim. For a more realistic behaviour it is also advisable to adjust the restitution of the material. The coefficient of restitution represents the bounciness or elasticity of objects, determining the extent to which they bounce or rebound off each other after a collision. In this case it is set to 0.3 for PLA.

4.3 Control

The control of this setup is centralised in the simulation environment (Isaac Sim), which is the brain in charge of communicating with the rest of the elements using different APIs, libraries for high-level planning and pre-established orders that it has to follow to control the devices involved in the experiment, both in the simulation and in the physical setup,as illustrated in Figure 4.18.

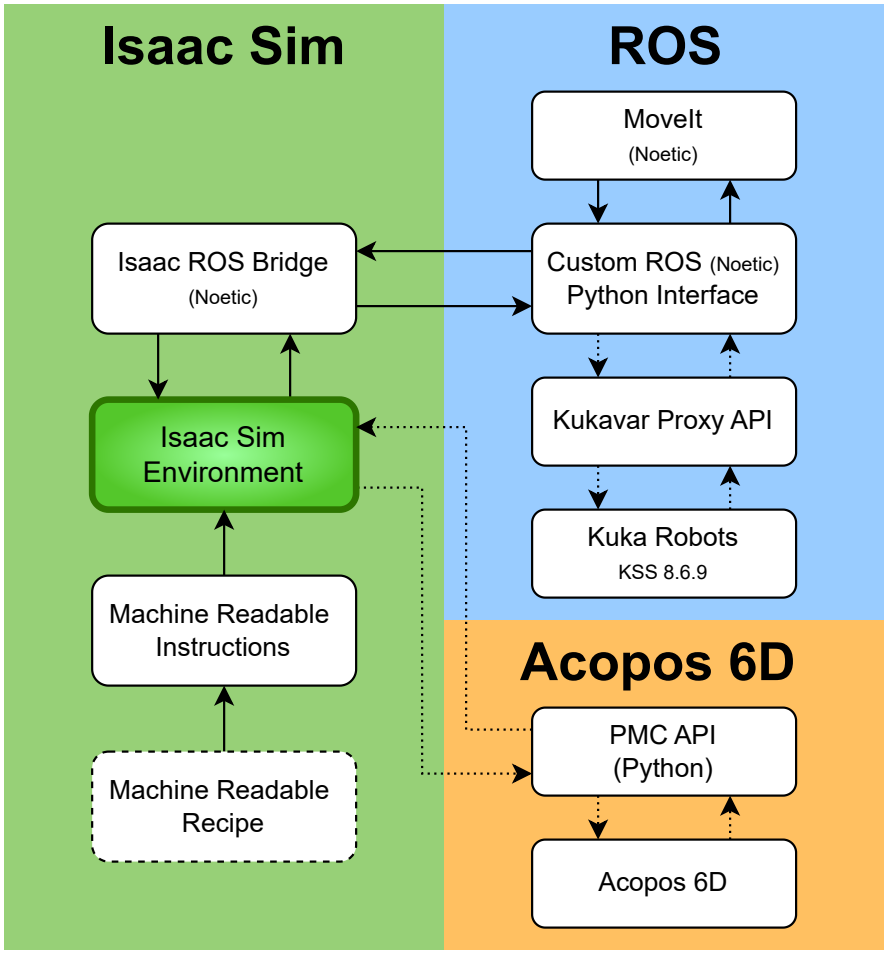

Figure 4.18: Communication workflow between ROS and physical robots (blue), Simulation environment (green) and Magnetic levitation platform (orange). *Machine Readable Recipe* is not implemented.

In the first place, the simulation will read out a series of actions that must be followed sequentially. These actions are low level, i.e. a shuttle is given the command to move to such a position or a certain gripper is given the command to open. However, the idea is to implement a high-level recipe interpreter that, given "human" commands such as "stir the solution for 2 hours" (this syntax is used for text to procedure option in RXN subsubsection 2.2.2), automatically creates the commands to move the shuttle to the stirring station, so the robot picks up the tray and puts it on the stirring machine until the set time has passed and finally returns the tray to the platform.

Once the roadmap for the experiment is established, the experiment is run from the simulation environment. The implemented control includes 2 modes, simulation only or digital twin. In the case of simulation only, the interface with the physical robots is skipped, In the case of simulation only, the interface with the physical robots is dispensed with, but MoveIt is still used to plan the robot trajectories and a simple algorithm within the simulation is used to drive the shuttles. When using the digital twin mode, the PMC API is used to control the platforms, dispensing with the control implemented in simulation and using the Kukavar interface to send and read the trajectories of the robotic arms.

4.3.1 Robot Operating System

Robot Operating System (ROS) [61] is an open-source platform developed with the purpose of integrating libraries and tools seamlessly with different robotic systems. ROS functions by allowing for the creation of different topics in which applications can publish and/or subscribe (listen) to specific topic containing data, allowing for easier integration between different applications. In this project, ROS is used to establish a communication between Nvidia Isaac Sim (simulation) and the Kuka manipulators (real world).

An useful tool used together with ROS is the MoveIt [62] library that is capable of, among others, motion planning, manipulation and collision avoidance for robotic applications. The MoveIt library has a configuration wizard named Moveit! Setup Assistant (MSA), Figure 4.19, which provides an easy-to-use graphical user interface (GUI).

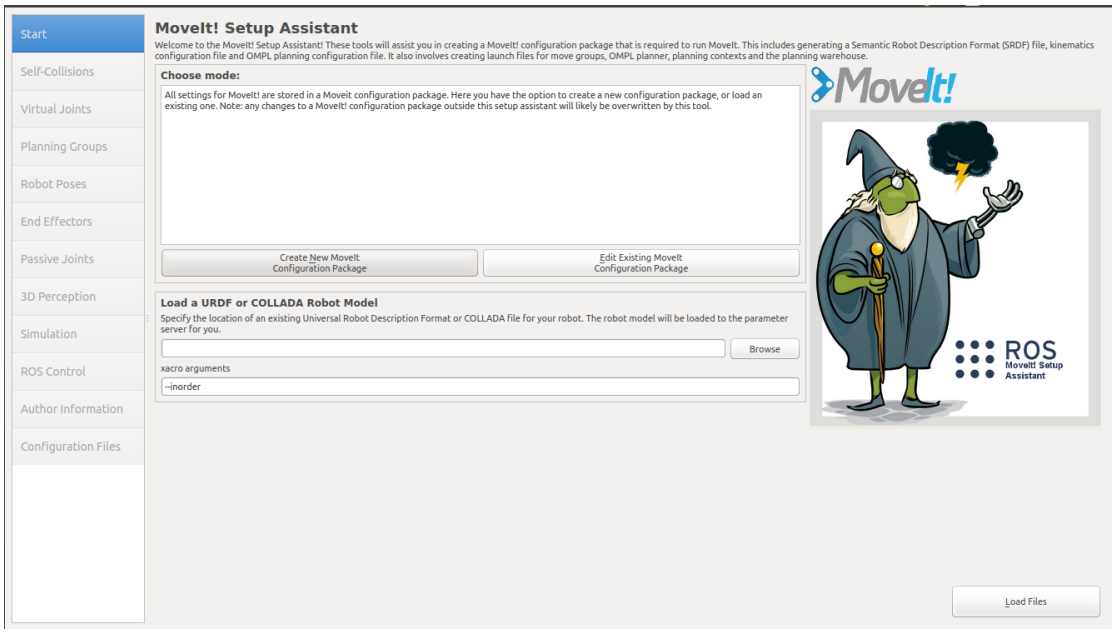

Figure 4.19: MoveIt Setup Assistant GUI.

MSA requires the URDF files of the robots to be controlled. In this project, the URDF files for both Kuka KR3 R540 and KR4 R600 were provided by ROS-Industrial [63], an open-source platform dedicated for the implementation of industrial hardware in ROS. The URDF for the Schunk 25 NNB was created using the URDF export tool in Solidworks. Once the URDF is provided, the MSA is now able to generate self-collisions with all the present objects, create planning groups for the robots (including kinematics), create different poses for the defined planning groups, define the end effector and add controls for the planning groups. Once the setup is completed, all the mentioned parameters are added in a configuration file that can be used to control the robot in different environments, such as in Rviz, Isaac Sim and in the physical setup.

ROS is considered for this project since it is able to interface with both the simulation and the physical environments, which means that through ROS, Nvidia Isaac Sim can be connected to the physical setup. Moreover, using ROS will allow the usage of the Moveit packages in Nvidia Isaac Sim, as well as in the physical setup allowing for the same control method in both environments. There are different control methods that can be implemented in Nvidia Isaac Sim for controlling the Kuka manipulators, such as: RMPflow, Articulation Motion Policy or Path Planner, but none of them is capable of being transferred onto the physical manipulator. As for the real Kuka manipulator, it can be controlled via the build-in pendant or via a Programmable Logic Controller (PLC). Without the usage of ROS, establishing a communication between the simulation and the world can be a challenging task since different methods have to be properly synchronised and there is a chance for them to not correlate with each other.

ROS1 and ROS2

There are 2 variations of ROS, ROS1 and ROS2. Both variations work based on the same topic principle, but ROS2 uses Data Distribution Service (DDS), which is able to provide greater efficiency, reliability and real-time capabilities than its predecessor, ROS1, which uses a Master-Slave architecture and XML-RPC middleware [64].

Between ROS1 and ROS2 there are some advantages and disadvantages when considering the implementation. The main advantage of ROS1 is that most of the libraries and tools are well developed and widely available, while some ROS2 packages are still under development. At the same time, ROS2's advantage is that it will have a long term support, while ROS1 will become obsolete in 2025 [65].

While considering the implementation of ROS, ROS1 has an advantage due to the well developed packages and tools which will ease the implementation, but alternatively ROS2, will offer a much longer support of the current project. In order to make a choice, a sprint session will be conducted for both ROS1 and ROS2, starting with ROS2.

ROS2 Sprint session

Since the focus was on a future-proofed platform, ROS2 was considered the first one for the sprint session, and if ROS2 was viable, ROS1 would have been discarded automatically. Similar to any other ROS, ROS2 is available on different Ubuntu versions, Ubuntu 20 (ROS2 Foxy) and Ubuntu 22 (ROS2 Humble). Since Isaac Sim recommends ROS2 Foxy due to the ROS2 bridge, which at the time was fully released for ROS2 Foxy and in beta for ROS2 Humble, the choice was to move forward using ROS2 Foxy. After some investigation it was found that the MSA is not available in ROS2 Foxy, but only in ROS2 Humble and a workaround can be achieved by using Docker to create a ROS2 Humble image on which MSA can be installed and used to create the necessary configuration files. Based on this information, it was decided to continue the sprint session and investigate the usage of Docker and ROS2 Humble.

Once ROS2 Humble was installed in Docker, next step was to create the configuration files for the Kuka robots. Since the support packages were made for ROS1, they had to be manually changed to work in ROS2, mainly the CMake files. Once the support packages were successfully converted, they were loaded in MSA and the configuration files were made.

The next step was to take the configuration files made in Docker and add them in the ROS2 Foxy local workspace and test them using Rviz (vizualization/debugging tool for the MSA configuration files). Unfortunately, the configuration files created in ROS2 Humble are not compatible with ROS2 Foxy, because the Moveit package is missing an essential library, *moveit_configs_utils*. It was possible to change the launch files in the MSA configuration folder, so they can be loaded in Rviz, but the controllers could not be loaded and the robot was not able to move.

At this point of the sprint session, it was clear that ROS2 will require a lot of time due to all the incompatibilities and shortcomings of each version, ROS2 Foxy not having MSA and ROS2 Humble being in beta for Isaac Sim.

ROS1 Sprint session

ROS1 Noetic (Ubuntu 20) was chosen for the ROS1 sprint session since it is the recommended version by Isaac Sim. Due to the maturity of ROS1 and because all the required packages were made with ROS1 in mind, the implementation did not require any workaround, such as using Docker, and it was notably easier and less time consuming compared to ROS2 and it was decided to continue on the ROS1 platform. ROS2 can be considered once again when Nvidia Isaac Sim will have a stable release for their ROS2 bridge supporting Humble, which will allow for the usage of MSA natively.

Control Multiple Robots with MoveIt

Once ROS1 was chosen, the next step was to create the configuration files for Kuka KR3 R540 and Kuka KR4 R600 using MoveIt. Since MoveIt is able to create collision free path planning for the robots, the whole system had to be loaded, which includes the 5 Kuka manipulators and the Acopos 6D table, so MoveIt is able to determine any possible collision with the environment.

The initial development consisted of having a MoveIt config for each type of robot together with an action graph in Isaac Sim independent for each robot within an assigned ROS namespace. This solution provided great versatility to the project since each arm could be moved, added, removed or replaced by another model by modifying just a few lines of code.

Nevertheless, *MoveGroupCommander*, that is the MoveIt function (from MoveIt python API) in charge of make the planning, is not able to handle several instances at the same time, meaning that is not possible to command each *move group* (robotic arm) under different namespaces simultaneously.

The official solution provided by MoveIt to control several robots is to load them in the same URDF file and including each robot in a different planning group. Although this method makes the setup creation process more tedious, it provides a really valuable advantage, it allows to consider the position of the rest of the robotic arms in the planning, that is, it provides collision avoidance being a much safer way of working. Furthermore, by including the assets that will be present in the simulated environment, these are considered for collision avoidance.

A xacro file is created to facilitate the creation of the environment. This calls the macros of the different elements. Therefore, the URDF file is created by calling the xacro macro of each robot and its corresponding gripper in the required position, adding it to the configuration. The URDF file contains the following assets:

- 2x Kuka KR3 R540 with gripper
- 1x Kuka KR4 R600 with gripper
- 1x Kuka KR3 R540 with solid dispenser
- 1x Kuka KR3 R540 with pipette (dispensing/aspiration)
- 1x Acopos 6D table (including the flyways)

To interact with the simulation, a launch file is executed in parallel. This file initialises RViz and launches a python script in charge of managing the communication between the different ROS nodes. The script creates a ROS node (*kuka_combined_joints_publisher*) and is in charge of subscribing the request command topics (planning group as String and desired movement as JointState/Pose/PoseArray) coming from the Isaac MAPs extension node (*isaac_joint_request_publisher*) which are sent to *MoveGroupCommander* and is also subscirbing */joint_command_desired* topic to send the movement back to the simulation.

4.3.2 Planar Motor Controller

The Planar Motor Controller (PMC) is supplied with a Windows application (Planar Motor Tool) to work directly with the shuttles (also called Xbots). Furthermore, a Python API provided by Planar Motor will be used to interface with the PMC via TCP/IP to control the different Xbots and receive the exact position in real time, among other data, of each of them. By using this API it is possible to recreate the control done in the simulation directly on the real setup and viceversa, thereby having a digital twin of the Acopos 6D platform. The characteristics of the PMC can be seen in Table 4.1.

Characteristics	**Value**
Max Speed	3 m/s
Max Acceleration	> 20 m/s^2
Repeatability	< 5 micron
Levitation Height	0.4 - 4 mm
Flyway Dimensions	240 x 240 x 70 mm^3
Wash Down	Yes
Auto Routing	Yes
Collision Avoidance	Built-in

Table 4.1: Planar Motor Controller Specifications. Accessed: 14/04/2023, [66].

Among the functions provided by the API are the system functions, which allow to connect to the device via TCP/IP, check the device status, reboot the PMC or gain master control. On the other hand, the control functions are extensive, those used in this project include activating xbots, getting the status of xbots (individually or as a whole), moving an xbot to a given position, moving all trays at once to their given targets, and recovering an xbot that has lost connection. Nonetheless, the API offers more functions such as rotating movement of trays, macros for sequences of movements and more functions.

4.3.3 Orchestration

The orchestration of the experiment is one of the cornerstones for the fully autonomous operation of a laboratory. Therefore, its integration has been taken into account in the design and implementation of the entire workflow. Furthermore, several visits have been made to observe and understand how the chemists perform their experiments, analysing the steps to carry out a synthesis experiment and the key aspects to optimise section 2.4.

Additionally, it is considered that for this implementation to be a scalable solution, it should not be over-fitted to a single existing orchestrator. As our approach also takes into account the actual implementation in the available configuration, the work has focused on creating a scalable low-level interface. To achieve this, the experiment instructions are passed in a *.yaml* file and are intuitively modifiable, as shown in Figure 4.20. These instructions are executed from the simulation sequentially, checking when an action has finished in order to start the next one.

```yaml
- action: MOVE_TO_JOINT_STATE # Move to platform (BEAKER)
  parameters:
    planning_group: 'robot_arm_5'
    joint_state_request: [ 0.4951248, -0.18098253, 2.1004589, 3.140828,
    ↪  -0.65211844, -1.0741844 ]

- action: MOVE_TO_POSE
  parameters:
    planning_group: 'robot_arm_1'
    position: [0.855, -0.140, 0.30]
    orientation: [0.0, 1.5708, 0.0]

- action: MOVE_ALONG_CARTESIAN_PATH
  parameters:
    planning_group: 'robot_arm_2'
    waypoints:
      - [[1.0, 2.0, 0.45], [0.0, 1.5708, 0.0]]
      - [[1.5, 2.5, 0.273], [0.0, 1.5708, 0.0]]

- action: MOVE_SHUTTLE_TO_TARGET
  parameters:
    xbot_id: 1
    target_x: 5
    target_y: 5

- action: GRIPPER_CONTROL
  parameters:
    planning_group: 'robot_hand_1'
    state: 'open'
```

Figure 4.20: Example of some of the Low-level Orchestration Commands in YAML

This set of commands provides what is needed to perform more complex tasks, such as picking up a tray on a given shuttle to place it on the stirring machine or dispense a specific amount of compound into a specific beaker. The RXN "Text to Procedure" application (subsubsection 2.2.2) provides a high-level interface for experiment orchestration. As shown in Figure 4.21, it is only needed a text description of an experiment to obtain the necessary instructions to carry it out.

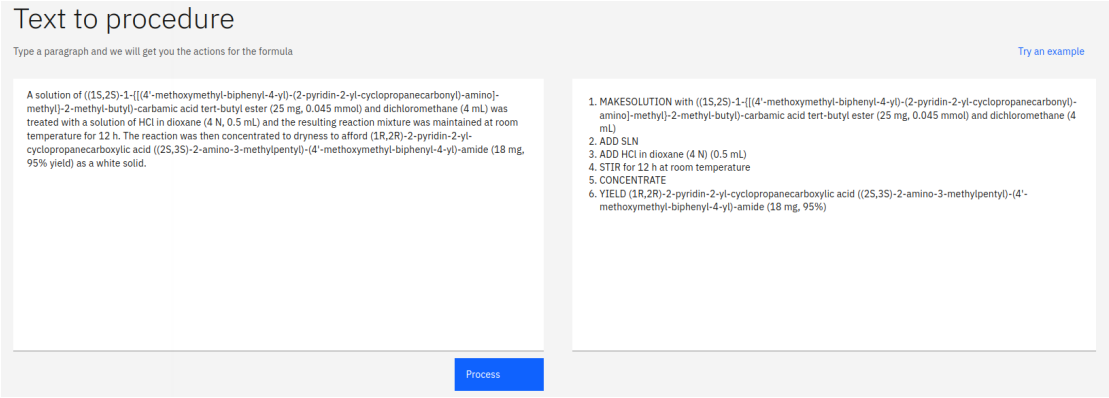

Figure 4.21: Text to procedure feature from RXN application.

Based on this approach, it will be possible to translate these commands into instructions interpretable by the self-driven laboratory, taking into account the current setup (i.e. the tasks it can carry out), the available stock and other relevant parameters. It is also worth mentioning that there are other orchestrators, as mentioned in the subsection 2.2.2, that can serve as brains to greatly optimize the performance of the MAP.

4.4 Physical Setup

The AAU Smart Production Laboratory currently has a fully functional installation consisting of 5 Kuka robots and a Acopos 6D platform. There are two different models of robotic arms arranged around the platform, 4 of them are model KR3 R540 and the other one is a model slightly larger, KR4 R600. Table 4.2 shows their specifications, and photo Figure 4.22 shows their distribution. The installed B&R magnetic levitation platform consists of 3x4 flyways of 24x24 cm, and 8 shuttles of 12x12 cm, as shown in picture Figure 4.23. Moreover, the setup includes a planar motor controller, explained in subsection 4.3.2.

Characteristics	Kuka KR3 R540	Kuka KR4 R600
Axes	6	6
Payload	3.0 kg	4.6 kg
Reach	541 mm	601 mm
Repeatability	0.02 mm	0.015 mm
Robot Mass	26 kg	24 kg
Controller version	8.6.9	8.6.9

Table 4.2: Specifications of the Kuka robot arms.

Figure 4.22: AAU Matrix Production Setup. The kuka manipulator arms used can be observed. Kuka 5 is the only Kuka KR4 R600, the others are Kuka KR3 R540. The red and green arrows show the x and y axes of the Acopos 6D platform respectively, which is composed of 12 flyways.

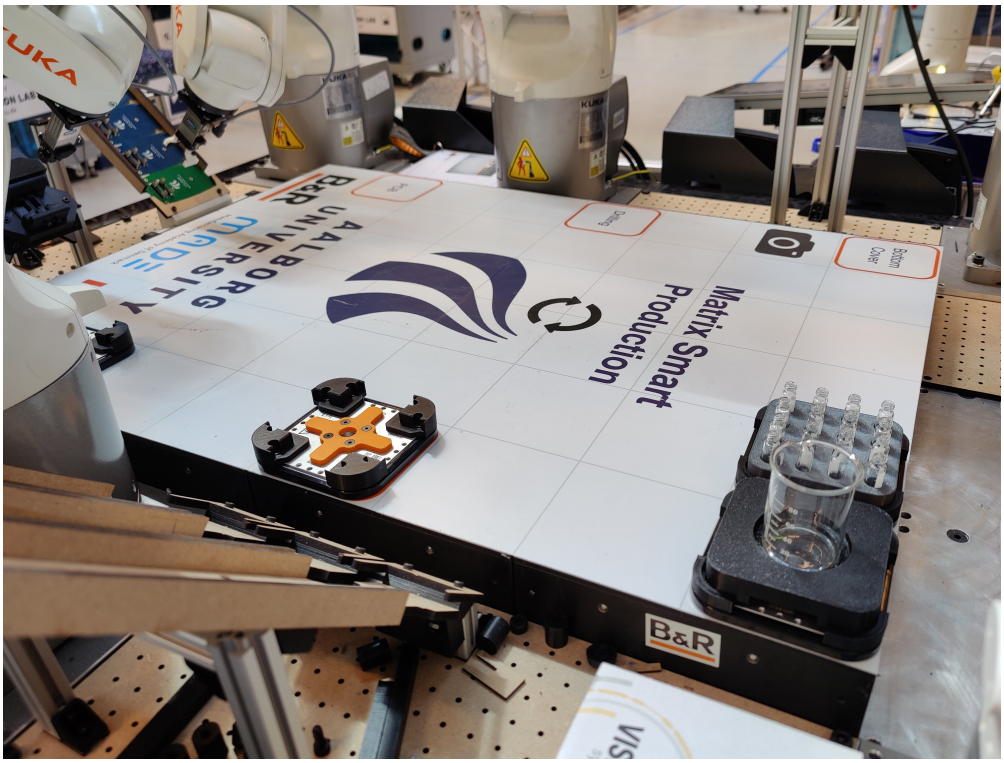

Figure 4.23: Acopos 6D magnetic levitation platform. The shuttles with the beaker and vials tray can be observed.

4.4.1 Computer-Aided Design

Since the AAU Smart Production Laboratory was designed for a different application, some new parts had to be custom made. Since the MAP setup is in a prototype stage, 3D printing the new parts was the fastest, safest and cheapest option. Moreover, the entire Isaac environment requires some form of CAD models (Figure 4.24, resulting in a need for 3D modelling in both setups, digital and real.

Most of the parts were already created and made available by previous groups that engaged in the AAU Smart Production project, such as the Schunk 25 NNB gripper and the Acopos 6D system (including the shuttles). The additional parts needed for the MAP project were:

- Fixtures for holding the trays on the shuttles (4 fixtures/shuttle)
- Trays for holding HPLC vials (16 vials/tray, 2mL vials)
- Trays for holding beakers (1 beaker/tray, 100, 200 or 300 mL beakers)
- Fingers for the Schunk grippers capable of lifting the trays while loaded
- Fake dispenser (testing purposes)
- Fake electronic pipette/aspirator (testing purposes)

Figure 4.24: CAD models (custom made)

Figure 4.25: Real models (3D printed)

4.4.2 ROS implementation on KUKA

The first step was to follow the implementation provided by ROS-Industrial which requires the Kuka's Robot Sensor Interface (RSI) software. The RSI versions are tied to the Kuka Software System (KSS) that is running on the controller. The Kuka controllers installed in the AAU Smart Production Laboratory have KSS version 8.6.9 which requires RSI version 4.1. The provided example by ROS-Industrial was made using RSI version 3 and it was found that the files are incompatible with RSI 4.1.

Kuka provides an Ethernet use example for RSI 4.1 including the necessary files and documentation in order to establish a communication between the controller and the PC. Once the files were loaded and the IP and Port numbers were changed, a new problem was found related to the communication speed. In the RSI documentation it is specified that the connection requires a ping of less than 12ms and if there is any delay, RSI will stop working. To address this problem, a PC running a real time kernel was used to connect to the controller, achieving a stable communication of less than 1ms ping, meaning that the RSI communication is stable at all times. Even though the communication was stable, when selecting and starting the program in the Kuka controller, it would be able to open the RSI communication and go to an arbitrarily home position, but when the program would reach the function that listens to the server (RSI_MOVECORR() function) the controller would give 2 errors.

First error, KS00489 "Stop by $Correction-functionality originator A2 (axis 2/joint 2)" which is believed that is caused by RSI receiving a too large correction. First attempt at solving the problem was to set joint 2 to 0° since the first command that RSI receives is to go to that joint angle, but this did not solved the problem. The second attempt was to check the RSI files that were provided by Kuka in their example. The files were loaded in WorkVisual (visualization and IDE software for Kuka's connections) for inspection, but there was not an apparent problem and the issue could not be fixed.

The second error, KSS29002 "Object ETHERNET1 returns error RSITimeout", originates from the RSI 4.1 software and, based on the provided documentation, an attempt at solving the error was made, but it was unsuccessful. It is believed that both errors are closely related because the documentation on error KS29002 mentions the correction mode.

Due to the inability to solve both errors, another solution had to be found and implemented so the Kuka can be controlled using ROS. While conducting some further research on ways to connect the Kuka robot to ROS, a solution called C3 Bridge [67], a plugin used by RoboDK (OLP software), was found that is using the Kuka Proxy Var, a TCP/IP solution made for communication between PC and the Kuka controller. It is working by creating a server and opening a communication port on the controller side to which an external PC can connect using the Ethernet connection. While investigating the Kuka Proxy Var, a ROS implementation using the Kuka Kr6r900sixx was found [68]. This solution had to be modified to work with the Kuka Kr3r540 and Kuka Kr4r600.

Chapter 5

Experiments and Results

This chapter provides an overview of the experiments conducted in both simulated and physical environments, together with documented results. Firstly, the experiment in simulation is presented along with the challenges encountered. Subsequently, the metrics for evaluating the performance of the experiment during the simulation are analysed. Lastly, the experiments carried out on the physical setup are discussed in more detail.

5.1 Simulation Experiment

Initially, the experiments are carried out within the Nvidia Isaac Sim simulation environment, based on the design depicted in Figure 4.4 - Figure 4.7. The simulation experiment aims to validate the successful execution of the experiment in an automated and sequential manner. This requires the orchestration of the experiment, which will coordinate the actions of the robotic arms with their corresponding grippers and the shuttles. To ensure the correct functioning of the orchestration process, the independent actions listed hereafter are tested:

- **Shuttles movement**
- **Robotic arms movement**
- **Pick and place of trays**
- **Shuttle and tray movement**

Once all these actions have been verified to work successfully, the correct functioning of the orchestration can be tested.

5.1.1 Shuttles movement

Requirement: The shuttle has to be able to move within the bounds of the table on both x and y axis.

Due to the requirement of a physical connection between the PMC library and the Acopos 6D platform to function, a workaround was devised to enable shuttle movement within the simulation environment without the need for a physical connection to the actual system. A dedicated function was developed, allowing the shuttles to be assigned a target position (x, y) and moved to their destination. The successful outcome of this test relies on the shuttles capability to accurately move and reach the specified target location.

Result: The shuttles demonstrate their ability to reach the target position with the given velocity, but they are not able to avoid collisions between each other.

5.1.2 Robotic arms movement

Requirement: The robots must be able to communicate with MoveIt and move to the desired positions, such as pick locations or place locations.

For this test, the URDF containing all the robots and the action graph presented in subsubsection 4.3.1 and subsubsection 4.2.2 respectively, were imported into the simulation environment. Once both parts are loaded within the environment, the configuration file made by MSA is launched. This configuration file is launched from a ROS launch file which, in turn, initialises a python script that allows direct communication between the MAPs script and MoveIt through different ROS topics.

Result: It was possible to control each robot in the simulation environment by using RViz or sending commands from the MAPs script. Due to the limitations of MoveIt described in subsubsection 4.3.1, still only one robot can be controlled at the same time, but this can be done under the same instance of MoveIt. Furthermore, collision between the robotic arms and the environment (included in the URDF file) is avoided.

5.1.3 Pick and place of trays

Requirement: The robots have to be able to pick and place the trays

The tray gripping test was done after the parameters presented in subsection 4.2.4 were chosen. The experiment was conducted by loading the simulated environment in Nvidia Isaac Sim and controlling one of the Kuka manipulators from Rviz, similar to how the robot movement test in subsection 5.1.2 was done. This process was done manually by sending commands from RViz to Nvidia Isaac Sim.

Result: The manipulator was able to pick up the tray and place it to a new location as shown in Figure 5.1.

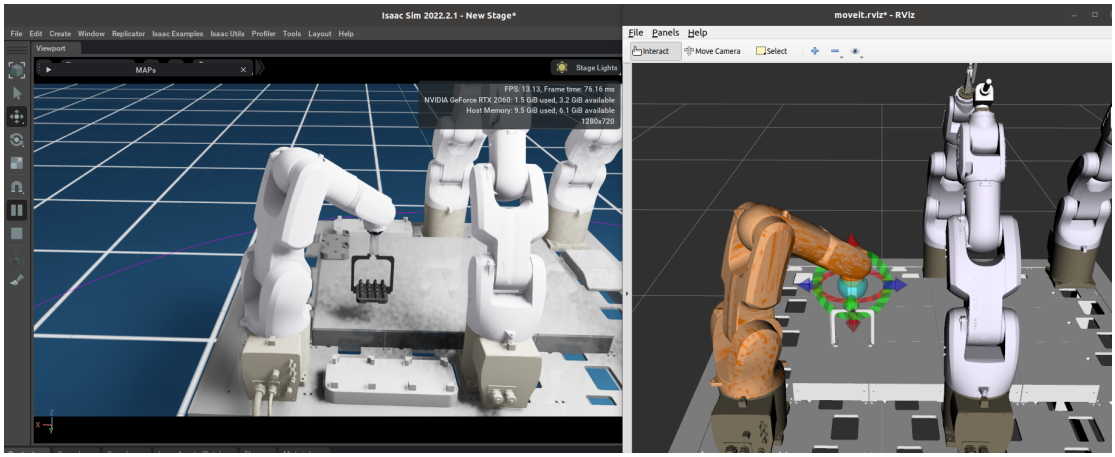

Figure 5.1: The Kuka robot lifting the tray containing the vials. The left side of the figure shows the Nvidia Isaac Sim environment while the right side shows Rviz. Picture taken during debugging.

5.1.4 Shuttle and tray movement

Requirement: Once the tray is placed on a shuttle, the shuttle has to be able to move to the destination with the tray on top.

An important part of the system is the ability to transport the trays from station to station using the shuttles and while in the physical world placing the tray on the shuttle and moving the shuttle together with the tray is a trivial task, in the simulation, due to the physics and colliders it can become a challenging task. The goal of this experiment is that once the tray is placed on the shuttle, the shuttle can move the tray to the next station.

Result: The tray is able to be moved by the shuttle. This is done by scripting the connection between the shuttle and the tray and when the shuttle moves, it translates the position to the tray.

5.1.5 Orchestration test

Requirement: The orchestration must command and complete all steps automatically, without human intervention.

Since all the individual parts of the simulation are functional, the orchestration test can be conducted. A .yaml file, similar to Figure 4.20, was created, where each action for the robots, grippers and shuttles was defined. The orchestration test will use the entire simulation environment as well as the ROS side of the project. The actions defined in the orchestration file are made based on the design workflow presented in Figure 4.1 and aims to recreated the experiment described in section 2.4.

Result: The simulation was able to automatically iterate through all the orchestration actions in the instructions recipe, resulting in a successful orchestration test.

The Figure 5.2 shows some of the actions mentioned during the conduct of a whole experiment.

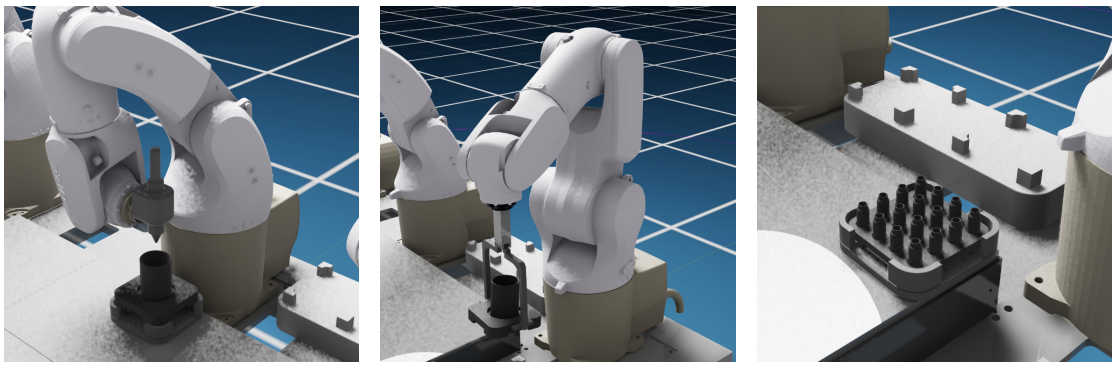

Figure 5.2: Screenshots of the experiment in simulation. From left to right, picking up and placing a tray, dispensing and moving the tray in the shuttle.

5.2 Simulation Results

To objectively analyse the performance of the experiment, data were collected during the simulation experiments. The data collected to evaluate the performance of the experiment are the number of actions performed by each individual work unit (robotic arms, grippers and shuttles) and the overall time that each of them works during an experiment. These data are presented in Table 5.1.

	Nº Actions	Time 1 (s)	Time 2 (s)	Time 3 (s)	Time 4 (s)	Time 5 (s)
Kuka arm 1	9	105.23	95.96	100.29	94.92	99.26
Kuka arm 2	7	116.565	110.24	113.63	111.55	113.69
Kuka arm 3	9	114.67	105.99	109.08	105.97	109.07
Kuka arm 4	3	56.13	50.77	51.75	50.89	51.2
Kuka arm 5	17	212.94	201.37	213.67	202.65	211.92
Gripper 1	4	12.47	16.12	13.55	11.48	14.12
Gripper 2	4	12.78	13.54	10.5	9.69	11.03
Gripper 5	6	21.14	24.92	18.45	19.98	22.29
Shuttle 1	7	23.78	23.09	26.03	24.91	22.01
Shuttle 2	4	11.924	11.1	11.26	11.08	11.04
Total	70	687.629	653.1	668.21	643.12	665.63

Table 5.1: Experiment performance in simulation for each unit. The table shows the number of actions executed for each unit in a single experiment and the overall working time for 5 equal experiments.

The Figure 5.3 illustrates the workload for each work unit. It is clear that the Kuka arm 5, which corresponds to the Kuka KR4 R600, has a workload nearly double that of the others, which is accentuated if the pick and place actions carried out by its corresponding gripper are taken into account.

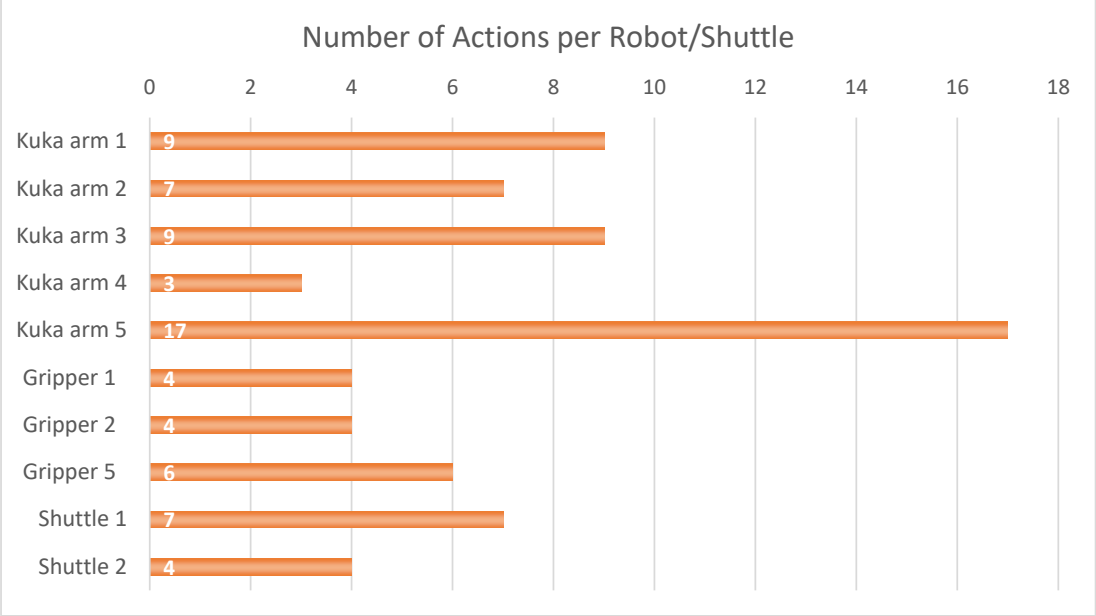

Figure 5.3: Number of actions executed for each work unit during the experiment process.

The experiment has a great potential for optimisation, both in the assignment of tasks and in the design of the experiment itself, however, it may be recalled that the experiment is carried out recreating the existing setup in the AAU laboratory, which offers great versatility but has not been designed for this particular task.

Nevertheless, the total average time of the experiment in simulation oscillates between 643 and 688 seconds, i.e. roughly from about 10:40 min to 11:30 min. To this time should be added the working times corresponding to the solids dispenser, the absorption and pipette dispensation and the mixing time in the stirring machine. Although there is no way to measure these times precisely, it is expected that the total time will still be much shorter than the same experiment carried out by humans as the one explained in section 2.4.

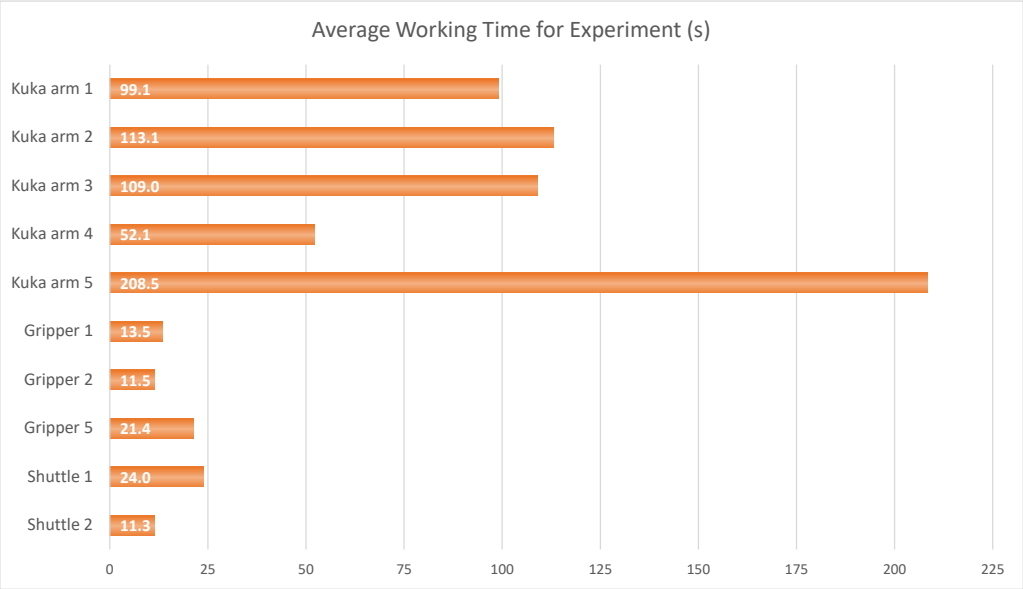

Figure 5.4: Average working time per experiment for each robotic arm, gripper or shuttle.The average time is obtained by adding the execution times of the actions performed by each device over 5 complete experiments.

In Figure 5.4 it can be seen that the time taken by the robotic arms is, in general, much longer than that required by the shuttle or grippers. This can be appreciated by comparing the working time of *kuka arm 2* with *gripper 5* and *shuttle 2*, which perform 7, 6 and 7 actions respectively.

The performance of robotic arms can be optimised in several ways. One of the factors that makes the arms more time consuming is that MoveIt is involved in every movement, i.e. every time a command is required to move, MoveIt has to do the trajectory planning, which takes extra time for each movement. Furthermore, by including the 5 robotic arms and a complex surrounding environment, the calculations are more extensive, but ensure that collisions between the arms or the surrounding environment will be avoided. Other factors such as the speed of movement of the robots and shuttles or the simulation run itself can affect the execution time.

5.3 Physical Setup Experiment

Once the simulation part of the experiments is done, the physical setup experiments will be conducted were the communication between parts of the project will be tested as well as the entire orchestration. For testing the physical setup orchestration, the following parts have to be working:

- **Acopos 6D control**
- **Robotics arms control**

When the individual parts are working, the orchestration test can be conducted.

5.3.1 Acopos 6D Control

Requirement: Nvidia Isaac Sim has to be able to communicate with the Acopos 6D platform in order to send and receive the shuttle's positions.

As explained in section 4.3, an API is used to communicate with the PMC. During the operation with the real setup, only the values of the shuttles in the real setup are used, however, the movement commands are sent from the MAPs script.

Result: Since the shuttles can be moved directly from Nvidia Isaac Sim, the test is considered successful. Moreover, the communication between the Nvidia Isaac Sim and Acopos 6D is able to transmit data in both ways, and because the Nvidia Isaac Sim is a representation of the real setup, the system can also be considered to be a digital twin of the Acopos 6D platform.

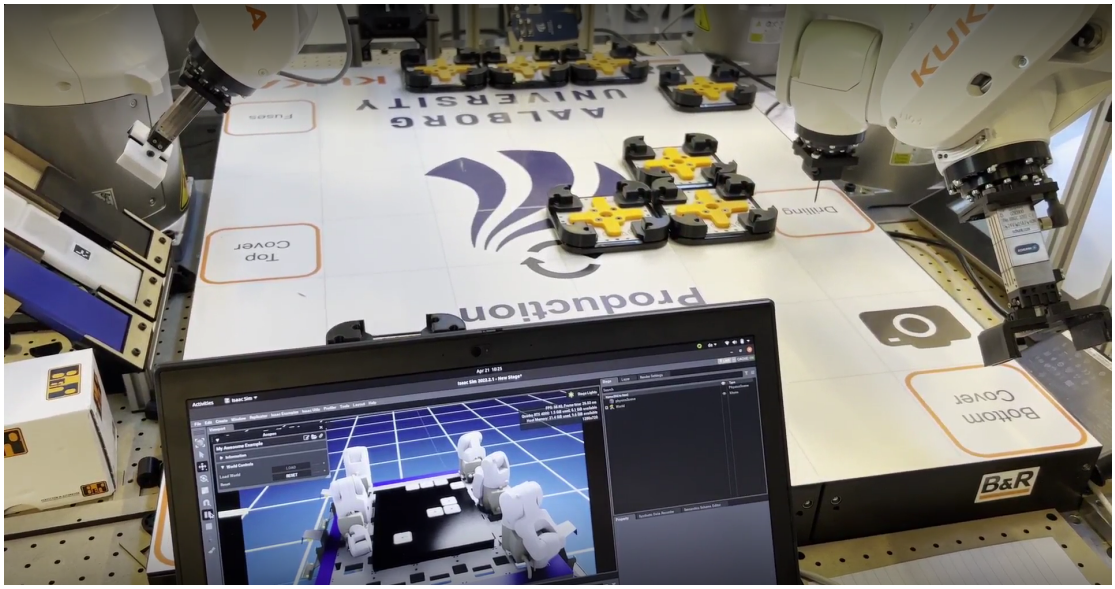

Figure 5.5: Testing the control of the Acopos 6D platform from the simulation environment in Isaac Sim.

5.3.2 Robotic Arms Control

Requirement: All robot manipulators must be able to communicate via ROS simultaneously with the simulation environment. Grippers are also considered part of the robot control.

The testing started by trying to implement the Kuka Var Proxy communication, presented in subsection 4.4.2, on one robot at a time. This is done by changing the IP address in the ROS launch file or by adding the IP number as an argument in the launch command to target the desired robot.

Result: Upon testing, the Kuka Var Proxy was able to communicate with the ROS hardware interface, hence it was possible to use ROS to control the real Kuka. Moreover, by adding the robot IP argument in the launch command, it was possible to connect to any robot present in the system. Unfortunately, this approach does not allow for establishing a communication with multiple robots at the same time, which would be required for the system to be fully functional and autonomous. Moreover, the robot movements during the experiments were jerky and different frequency rates were considered, but did not change the outcome.

Issue: The Schunk 25NNB grippers can not be used with the current setup, because there are no ROS controllers. Other possible solutions are to use the installed PLC or to purchase an I/O module for each robotic arm.

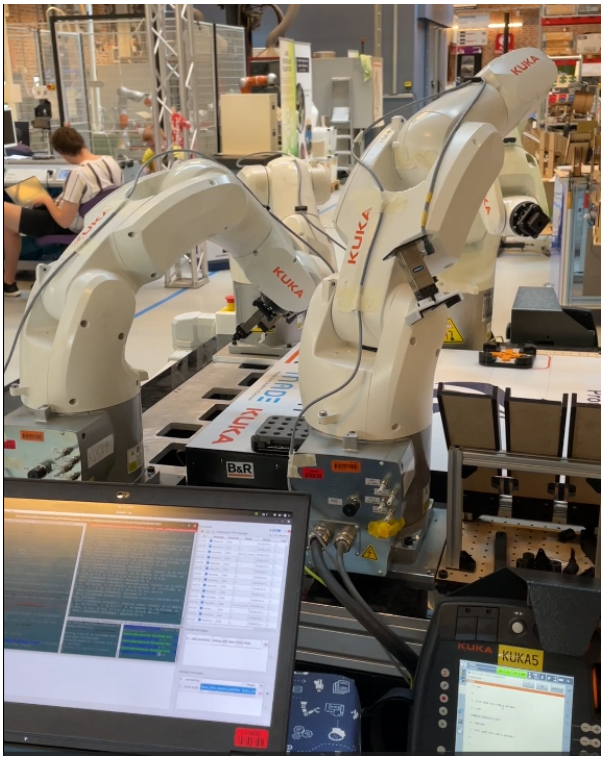

Figure 5.6: Picture taken during the communication test between ROS and Kuka Kr3r540 robot.

5.3.3 Orchestration test

Requirement: The orchestration must command and complete all steps automatically, without human intervention.

The orchestration for the physical setup is done in the same way as in the simulation (subsection 5.1.5) and the commands can be reused. However, it is assumed that there may be slight changes between the simulation and the real world and therefore a calibration would be necessary for it to perform properly.

Result: Due to the inability of establishing a communication with all the robots at the same time, and because the system is missing a communication protocol with the Schunk grippers, the orchestration test was not conducted. Even though all the robots can be controlled by changing the IP address, this approach requires the ROS hardware interface to be restarted in order to target a different robot.

Chapter 6
Discussion

In this chapter, the design will be evaluated together with the experiments and results presented in the preceding chapter. Subsequently, reflections on the global solution are presented, considering improvements and modifications that could have been incorporated. Finally, the chapter introduces a discussion of possible avenues for future research and development of this project.

6.1 Evaluation of Experiment Results

As explained in chapter 3, two sets of objectives are established, ones referring to the design, and others concerning the project solution.

Regarding design requirements, stated in subsection 3.3.1, these are contrasted with the design of the system realised in section 4.1. The essential requirements are fully satisfied, albeit some of the elements present are not real, such as the pipette or the dispenser, but they are valid for our proof of concept purposes. Regarding the extra requirements necessary to enable the setup to perform a multitude of different experiments, these are not considered in the main design based on the AAU Matrix Production, but are considered in the modular design proposed in subsubsection 4.1.1.

Hereafter, the results of the experiments described in chapter 5 are assessed based on the solution requirement criteria specified in subsection 3.3.2.

S1: The solution must be able to recreate the real setup in simulation
This requirement is fulfilled, all the necessary components of the real setup can be found by loading them from the MAPs extension created in Nvidia Isaac Sim, which can serve as a basis for future projects using the AAU Matrix Production setup. In the simulation environment, some realistic details such as textures, colours and backgrounds have been omitted as this increases the computational requirements.

S2: The solution must be modular
This requirement is not demonstrated in experiments, but is theoretically fulfilled. All the elements of the setup are easily modifiable from the MAPs script, except for the robotic arms and the table, which are more tedious to customise. If ROS is used to control several robots, it would require modifying the xacro files and creating a new MoveIt config to modify the setup.

S3: The solution must be able to perform an experiment in the real setup
The requirement has been partially fulfilled, a complete experiment in the real setup has not been achieved, but a lot of significant progress has been made along the way. It was possible to control the robots one at a time, as well as to control the Acopos 6D platform as a digital twin, and all the 3D printed elements required for the experiment were available. It was not possible to control all the robots together or to control the grippers.

S4: The solution must be able to carry out an experiment given certain instructions
This requirement is fulfilled, as shown in chapter 5 the whole experiment in simulation is carried out using the experiment orchestration consisting of machine-readable instructions. Likewise, this orchestration would be valid for an experiment in the physical setup.

S5: The solution must integrate AI-orchestration for the planning and execution of experiments
This requirement is not fulfilled, the presented solution is not able to directly integrate AI-orchestration for the planning and execution of experiments. Nonetheless, some progress has been made in that regard, such as the implementation of low-level orchestration using machine readable instructions that could serve as a gateway to interact with AI experiment orchestrators such as RXN.

Altogether, it can be concluded that the solution satisfies most of the requirements. Among the objectives that have not been fully completed, it is known that these were indeed ambitious, nevertheless, since this project can serve as a starting point for other implementations in the future, priority has been given to creating a solution to build upon rather than a very specific solution for the project application. This is covered in the following section 6.2

6.2 Reflections

The aim of this project is to redesign chemistry laboratories to accelerate the discovery of new materials using simulation and robotics. To this end, a functional proof-of-concept is intended to demonstrate this concept. Since the Smart Lab Production Lab at AAU will be conducting research in this field for the next few years, we are provided with a functional setup that fits reasonably well with our project but has not been designed for it, therefore the design decisions are not convenient for our purpose.

It should be noted that this provides the advantage that the physical setup is already installed and practically ready to go, but on the other hand, it limits the possible choices in the setup. As a result, the Kuka robotic arms and their corresponding Schunk grippers were already installed, as well as the Acopos 6D platform and the PMC.

It was decided to move forward without modifying the setup, which posed challenges such as controlling the robots using ROS and MoveIt without official support from Kuka, adapting the fingers of grippers with such a small stroke, or controlling the grippers without an I/O module available in the Kuka arms.

A crucial aspect to carry out the experiment in the physical setup was to establish communication with all Kuka robots simultaneously. It was possible to control the robotic arms one at a time, but this is not enough to carry out the experiment in the physical setup. To enable multiple connections concurrently, it is essential to modify the Kuka hardware interface mentioned in subsection 4.4.2.

As mentioned in the subsection 5.3.2, the robot presents some jerky movements that are believed to be due to the communication frequency, but changing the frequency rate did not improve the movement. Another option that has to be looked into is to use an Ubuntu that is running on a real-time kernel. Such a system is already available since it was used to debug the RSI communication between ROS and Kuka. The real-time kernel could potentially improve the communication with the Kuka server by giving real-time priority to the application.

Another critical aspect for the execution of the experiment was the implementation of the control of the physical grippers, which allows the robots to interact with the trays. As indicated in subsection 5.3.2, two possible approaches were identified: one using a programmable logic controller (PLC) and the other using the acquisition of an input/output (I/O) module. Both approaches pose different challenges.

In the case that using a PLC is chosen, it has the capability to control all the grippers and possibly a real pipette and dispenser, implying that only one PLC would be required, which is already integrated in the Matrix Production AAU. The main obstacle related to this strategy is caused by the simultaneous operation of ROS and PLC. Both frameworks would need a communication channel between them, which would increase the complexity of the project. The added complexity arises from the need for ROS, Nvidia Isaac Sim, PMC, and the PLC, to effectively communicate with each other.

Alternatively, the solution involving the use of I/O modules would involve the purchase and installation of an I/O module for each Kuka manipulator. Despite the possibility of a relatively less complex implementation than the PLC option, the steps for establishing communication with the I/O module have not yet been thoroughly investigated, so it is difficult to decide definitively which solution would best suit this project at this juncture.

The challenge of adapting the gripper fingers with such a small stroke was solved by designing and 3D printing fingers with a specific shape to fit with a hollow designed in the trays. However, the small stroke of the grippers hindered the work, requiring a very high precision for pick and place.

In order to be able to conduct diverse experiments that help to rapidly explore the vast space of possibilities in materials discovery, self-driving labs must be endowed with a high degree of versatility and consequently the orchestration that considers the existing system. This high-level orchestration must encompass the purpose of each workstation, the position and destination of the shuttles and the required movements of the manipulators.

Currently, a low-level orchestration is implemented, which requires the sequential definition of each shuttle and robot movement. A more convenient system would consist of a general experiment orchestration that, knowing the capabilities of the existing system (such as existing stations, possible actions, position, states) as well as stock of materials and given an experiment recipe, coordinates all the necessary actions to carry out the experiment. The creation of a high-level orchestration will pave the way towards the application of AI in orchestration. Once orchestration can operate based on system characteristics, an AI orchestration can be developed that, based on a chemical process and existing stock, autonomously selects the necessary stations and sequences. The implementation of AI-orchestration would allow the highest level of system autonomy. This goal was ambitious and could not be achieved mainly due to time constraints, however, a low-level orchestration has been achieved that can serve as a basis for future systems equipped with more autonomy.

6.3 Future Work

Since the experiments in the simulated environment were successful, more work has to be done towards the physical setup's success. The first task for having a successful implementation on the physical setup is to solve the problems explained in section 6.2 related to the concurrent communication with the robots and the use of the grippers. After these problems are solved, the physical system will require a real pipette and dispense device in order to conduct a real chemical experiment.

One future work that should be considered is to extend the Digital Twin concept to the entire system, since it is already working with the shuttle as presented in subsection 5.3.1. Implementing a Digital Twin for the entire platform will help visualise and monitor the metrics of the system at any time.

Based on the simulation results presented in section 5.2 it is visible that some Kuka arms are doing more actions that the others, hence it is believed that some level of optimisation could be achieved. A future work for this project should focus on analysing the results of different setups to further accelerated the processes.

Chapter 7

Conclusion

Chemical processes are and will still be a major part towards the discovery of new materials specifically tailored for different applications. As of right now, a lot of these processes are still still carried out by human workers, which can be slow and unreliable, and moreover, and laboratories are not designed to accommodate robotic solutions. As robotics can drastically increase the speed at which these processes can be performed as well as the reliability of the experiments by removing the human error, which impacts the repeatability of the experiment, this project aims at answering the following question:

How can the chemistry laboratories be redesigned by using simulation, robotic manipulators and magnetic levitation platforms with the purpose of drastically accelerating the discovery of new materials?

This project focused on answering the above stated question by converting the AAU Matrix Production setup into a self driven laboratory, using 5 Kuka manipulators and the Acopos 6D platform developed by B&R Automation. Different parts were designed in order to facilitate the experimental processes, in terms of transportation and handling. The solution was fully developed using a simulated environment in Nvidia Isaac Sim where the physical setup was recreated. ROS1 was used to control the robots, as it provided the capability to control both real and simulated robots. The shuttles were controlled by sending (x,y) position in the simulation and by using the PMC library for the real system.

Regarding the simulation side of the project, all the features are implemented and able to work together by following an orchestrator that sends different driving commands to the robots and shuttles. By following this orchestration, the system is able to conduct a fully automatic chemical experiment. Connecting to the physical setup proved to be a challenging task that has to be further investigated. The solution is able to control the physical shuttles and each Kuka robot, but it is not able to do it simultaneously. In view of these results, both in the simulation and in the physical setup, it can be concluded that robotic solutions are capable of taking over chemical processes and accelerating the discovery of new materials.

Bibliography

[1] Peter J. T. Morris. “The history of chemical laboratories: a thematic approach”. In: (2021). DOI: https://doi.org/10.1007/s40828-021-00146-x.

[2] Alán Aspuru-Guzik and Kristin A. Persson. “Materials Acceleration Platform: Accelerating Advanced Energy Materials Discovery by Integrating High-Throughput Methods and Artificial Intelligence.” In: 2018.

[3] Martin Seifrid et al. “Autonomous Chemical Experiments: Challenges and Perspectives on Establishing a Self-Driving Lab”. In: *Accounts of Chemical Research* 55.17 (2022). PMID: 35948428, pp. 2454–2466. DOI: 10.1021/acs.accounts.2c00220. eprint: https://doi.org/10.1021/acs.accounts.2c00220. URL: https://doi.org/10.1021/acs.accounts.2c00220.

[4] *CAPeX Pioneer Center for Accelerating P2X Materials Discovery*. https://www.dtu.dk/capex. Accessed: 2023-03-10.

[5] Martha M. Flores-Leonar et al. “Materials Acceleration Platforms: On the way to autonomous experimentation”. In: *Current Opinion in Green and Sustainable Chemistry* 25 (2020), p. 100370. ISSN: 2452-2236. DOI: https://doi.org/10.1016/j.cogsc.2020.100370. URL: https://www.sciencedirect.com/science/article/pii/S2452223620300596.

[6] Yunchao Xie et al. “Toward autonomous laboratories: Convergence of artificial intelligence and experimental automation”. In: *Progress in Materials Science* 132 (2023), p. 101043. ISSN: 0079-6425. DOI: https://doi.org/10.1016/j.pmatsci.2022.101043. URL: https://www.sciencedirect.com/science/article/pii/S0079642522001244.

[7] Melodie Christensen et al. “Automation isn’t automatic”. In: *Chem. Sci.* 12 (47 2021), pp. 15473–15490. DOI: 10.1039/D1SC04588A. URL: http://dx.doi.org/10.1039/D1SC04588A.

[8] Ivana Blaženović et al. “Comprehensive comparison of in silico MS/MS fragmentation tools of the CASMI contest: Database boosting is needed to achieve 93% accuracy”. In: *Journal of Cheminformatics* 9 (May 2017). DOI: 10.1186/s13321-017-0219-x.

[9] Nathan J. Szymanski et al. “Toward autonomous design and synthesis of novel inorganic materials”. In: *Mater. Horiz.* 8 (8 2021), pp. 2169–2198. DOI: 10.1039/D1MH00495F. URL: http://dx.doi.org/10.1039/D1MH00495F.

[10] Michael S. Bowen Robert W. Epps. “Artificial Chemist: An Autonomous Quantum Dot Synthesis Bot”. In: (2020). DOI: https://doi.org/10.1002/adma.202001626.

[11] Cheemspeed technologies. *Automated, parallel library synthesis*. 2020.

[12] F. G. L. PARLANE B. P. MACLEOD. “Self-driving laboratory for accelerated discovery of thin-film materials”. In: (2020). DOI: https://www.science.org/doi/10.1126/sciadv.aaz8867.

[13] Chenxu Xu Emory M. Chan. “Reproducible, High-Throughput Synthesis of Colloidal Nanocrystals for Optimization in Multidimensional Parameter Space”. In: (2010). DOI: https://doi.org/10.1021/nl100669s.

[14] Christian G. Berger Jerrit Wagner. “The evolution of Materials Acceleration Platforms: toward the laboratory of the future with AMANDA”. In: (2021). DOI: https://doi.org/10.1007/s10853-021-06281-7.

[15] LAURIE J. POINTS JONATHAN GRIZOU. “A curious formulation robot enables the discovery of a novel protocell behavior”. In: (2020). DOI: https://www.science.org/doi/10.1126/sciadv.aay4237.

[16] Phillip M. Maffettone Benjamin Burger1. “A mobile robotic chemist”. In: (2020). DOI: https://doi.org/10.1038/s41586-020-2442-2.

[17] Loïc M. Roch et al. “ChemOS: Orchestrating autonomous experimentation”. In: *Science Robotics* 3.19 (2018), eaat5559. DOI: 10.1126/scirobotics.aat5559. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.aat5559. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aat5559.

[18] Loïc M. Roch et al. “ChemOS: An orchestration software to democratize autonomous discovery”. In: *PLOS ONE* 15.4 (Apr. 2020), pp. 1–18. DOI: 10.1371/journal.pone.0229862. URL: https://doi.org/10.1371/journal.pone.0229862.

[19] Ian Pendleton et al. “Experiment Specification, Capture and Laboratory Automation Technology (ESCALATE): A software pipeline for automated chemical experimentation and data management”. In: *MRS Communications* 9 (June 2019), pp. 1–14. DOI: 10.1557/mrc.2019.72.

[20] Jiagen Li et al. “Toward “On-Demand” Materials Synthesis and Scientific Discovery through Intelligent Robots”. In: *Advanced Science* 7.7 (2020), p. 1901957. DOI: https://doi.org/10.1002/advs.201901957. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/advs.201901957. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/advs.201901957.

[21] Matthew L. Landry et al. “Simple Syntheses of CdSe Quantum Dots”. In: (2014). DOI: https://doi.org/10.1021/ed300568e.

[22] Philippe Schwaller et al. ""Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models". In: *Chem. Sci.* 9 (28 2018), pp. 6091–6098. DOI: 10.1039/C8SC02339E. URL: http://dx.doi.org/10.1039/C8SC02339E.

[23] Philippe Schwaller et al. "Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction". In: *ACS Central Science* 5.9 (2019). PMID: 31572784, pp. 1572–1583. DOI: 10.1021/acscentsci.9b00576. URL: https://doi.org/10.1021/acscentsci.9b00576.

[24] Philippe Schwaller et al. "Predicting retrosynthetic pathways using transformer-based models and a hyper-graph exploration strategy". In: *Chem. Sci.* 11 (12 2020), pp. 3316–3325. DOI: 10.1039/C9SC05704H. URL: http://dx.doi.org/10.1039/C9SC05704H.

[25] Alain C. Vaucher et al. "Automated extraction of chemical synthesis actions from experimental procedures". In: *Nature Communications* 11.1 (July 2020), p. 3601. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17266-6. URL: https://doi.org/10.1038/s41467-020-17266-6.

[26] Alain C. Vaucher et al. "Inferring experimental procedures from text-based representations of chemical reactions". In: *Nature Communications* 12.1 (May 2021), p. 2573. DOI: 10.1038/s41467-021-22951-1. URL: https://doi.org/10.1038/s41467-021-22951-1.

[27] *RXN for Chemistry*. https://rxn.app.accelerate.science/. Accessed: 2023-03-15.

[28] Florian Häse et al. "Phoenics: A Bayesian Optimizer for Chemistry". In: *ACS Central Science* 4.9 (2018). PMID: 30276246, pp. 1134–1145. DOI: 10.1021/acscentsci.8b00307. eprint: https://doi.org/10.1021/acscentsci.8b00307. URL: https://doi.org/10.1021/acscentsci.8b00307.

[29] Florian Häse et al. "Gryffin: An algorithm for Bayesian optimization of categorical variables informed by expert knowledge". In: *Applied Physics Reviews* 8.3 (2021), p. 031406. DOI: 10.1063/5.0048164. URL: https://doi.org/10.1063/5.0048164.

[30] Florian Häse, Loïc M. Roch, and Alán Aspuru-Guzik. "Chimera: enabling hierarchy based multi-objective optimization for self-driving laboratories". In: (2018). DOI: https://doi.org/10.1039/C8SC02239A.

[31] Riley J. Hickman et al. "Gemini: Dynamic Bias Correction for Autonomous Experimentation and Molecular Simulation". In: (2021). DOI: https://doi.org/10.48550/arXiv.2103.03391.

[32] Matteo Aldeghi et al. "Golem: An algorithm for robust experiment and process optimization". In: (2021). DOI: https://doi.org/10.1039/D1SC01545A.

[33] Stefan Langner et al. “Beyond Ternary OPV: High-Throughput Experimentation and Self-Driving Laboratories Optimize Multicomponent Systems”. In: *Advanced Materials* 32.14 (2020), p. 1907801. DOI: https://doi.org/10.1002/adma.201907801. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.201907801. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.201907801.

[34] Yang Bai et al. “Accelerated Discovery of Organic Polymer Photocatalysts for Hydrogen Evolution from Water through the Integration of Experiment and Theory”. In: *Journal of the American Chemical Society* 141.22 (2019). PMID: 31074272, pp. 9063–9071. DOI: 10.1021/jacs.9b03591. eprint: https://doi.org/10.1021/jacs.9b03591. URL: https://doi.org/10.1021/jacs.9b03591.

[35] Tony C Wu et al. “A Materials Acceleration Platform for Organic Laser Discovery”. In: *Advanced Materials* 35.6 (2023), p. 2207070. DOI: https://doi.org/10.1002/adma.202207070. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/adma.202207070. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/adma.202207070.

[36] Khalid Ridha Kadhim and Raghad Y. Mohammed. “Effect of Annealing Time on Structure, Morphology, and Optical Properties of Nanostructured CdO Thin Films Prepared by CBD Technique”. In: (2022). DOI: https://doi.org/10.3390/cryst12091315.

[37] Aksha Dhawan Suchita Singh. “Quantum Dots: An Emerging Tool for Point-of-Care Testing”. In: (2020). DOI: https://doi.org/10.1021/acsanm.0c01386.

[38] Monica A. Cotta. “Quantum Dots and Their Applications: What Lies Ahead?” In: (2020). DOI: https://doi.org/10.3390/mi11121058.

[39] Robert W. Epps Kameel Abdel-Latif. “Self-Driven Multistep Quantum Dot Synthesis Enabled by Autonomous Robotic Experimentation in Flow”. In: (2020). DOI: https://doi.org/10.1002/aisy.202000245.

[40] Gerasimos Konstantatos and Edward H. Sargent. “Nanostructured materials for photon detection”. In: (2010). DOI: https://doi.org/10.1038/nnano.2010.78.

[41] Guanying Chen et al. “Nanochemistry and nanomaterials for photovoltaics”. In: (2013). DOI: https://doi.org/10.1039/C3CS60054H.

[42] Parthiban Ramasamy et al. “Tunable, Bright, and Narrow-Band Luminescence from Colloidal Indium Phosphide Quantum Dots”. In: (2017). DOI: https://doi.org/10.1021/acs.chemmater.7b02204.

[43] Pedro E. M. Amaral et al. “Fibrous Phosphorus Quantum Dots for Cell Imaging”. In: (2020). DOI: https://doi.org/10.1021/acsanm.9b01786.

[44] Melanie Schranz et al. "Swarm Robotic Behaviors and Current Applications". In: *Frontiers in Robotics and AI* 7 (2020). ISSN: 2296-9144. DOI: 10.3389/frobt.2020.00036. URL: https://www.frontiersin.org/articles/10.3389/frobt.2020.00036.

[45] Ferdinando Auricchio. "A Continuous Model for the Simulation of Manufacturing Swarm Robotics". In: *Comput. Mech.* 70.1 (July 2022), pp. 155–162. ISSN: 0178-7675. DOI: 10.1007/s00466-022-02160-3. URL: https://doi.org/10.1007/s00466-022-02160-3.

[46] Gabriele Valentini, Eliseo Ferrante, and Marco Dorigo. "The Best-of-n Problem in Robot Swarms: Formalization, State of the Art, and Novel Perspectives". In: *Frontiers in Robotics and AI* 4 (2017). ISSN: 2296-9144. DOI: 10.3389/frobt.2017.00009. URL: https://www.frontiersin.org/articles/10.3389/frobt.2017.00009.

[47] Ahmad Reza Cheraghi, Sahdia Shahzad, and Kalman Graffi. *Past, Present, and Future of Swarm Robotics*. 2021. DOI: 10.48550/ARXIV.2101.00671. URL: https://arxiv.org/abs/2101.00671.

[48] Abid Haleem Mohd Javaid. "Enabling flexible manufacturing system (FMS) through the applications of industry 4.0 technologies." In: (2022). DOI: https://doi.org/10.1016/j.iotcps.2022.05.005.

[49] Dong Pang Junzheng Li. "A flexible manufacturing assembly system with deep reinforcementlearning." In: (2022). DOI: https://doi.org/10.1016/j.conengprac.2021.104957.

[50] G. Jayaprakash Mikell P. Groover. *Automation, Production Systems, and Computer-integrated Manufacturing*. 4th ed. 2015.

[51] Riku-Pekka Nikula et al. "Towards online adaptation of digital twins". In: (2020). DOI: https://doi.org/10.1515/eng-2020-0088.

[52] *Webots*. https://cyberbotics.com/. Accessed: 2023-05-15.

[53] *Gazebo*. https://gazebosim.org/home. Accessed: 2023-05-15.

[54] *Kuka Kr3r540 technical sheet*. https://www.kuka.com/-/media/kuka-downloads/imported/8350ff3ca11642998dbdc81dcc2ed44c/0000270971_en.pdf. Accessed: 2023-05-22.

[55] *Kuka Kr4r600 technical sheet*. https://www.kuka.com/-/media/kuka-downloads/imported/8350ff3ca11642998dbdc81dcc2ed44c/0000356043_en.pdf. Accessed: 2023-05-22.

[56] N. Roth and B. Schneider. "Exploring the performance of ROS2Clean Room Industrial Robot for Handling and Assembly in Semiconductor Industry". In: (1993). DOI: https://doi.org/10.1016/S0007-8506(07)62383-X.

[57] *Nvidia Isaac Sim*. https://docs.omniverse.nvidia.com/app_isaacsim/app_isaacsim/overview.html. Accessed: 2023-05-13.

[58] *Nvidia Omniverse*. https://www.nvidia.com/en-us/omniverse/. Accessed: 2023-05-13.

[59] *USD Pixar*. https://www.pixar.com/usd. Accessed: 2023-05-13.

[60] Şenol Şirin, Enes Aslan, and Gülşah Akincioğlu. "Effects of 3D-printed PLA material with different filling densities on coefficient of friction performance". In: (2023). DOI: https://doi.org/10.1108/RPJ-03-2022-0081.

[61] *Robotic Operating System*. https://www.ros.org/. Accessed: 2023-05-11.

[62] *MoveIt*. https://moveit.ros.org/. Accessed: 2023-05-12.

[63] *ROS-Industrial*. https://rosindustrial.org/. Accessed: 2023-05-11.

[64] Yuya Maruyama, Shinpei Kato, and Takuya Azumi. "Exploring the performance of ROS2". In: (2016). DOI: https://doi.org/10.1145/2968478.2968502.

[65] *ROS Distributions*. http://wiki.ros.org/Distributions. Accessed: 2023-05-11.

[66] *Planar Motor Systems*. https://www.planarmotor.com/en/products/. Accessed: 2023-04-14.

[67] *Kuka C3 Bridge*. https://robodk.com/doc/en/Robots-KUKA-RoboDK-driver-KUKA.html. Accessed: 2023-05-29.

[68] *Kuka Var Proxy implementation*. https://github.com/ImtsSrl/KUKAVARPROXY. Accessed: 2023-05-29.

Appendix A

MAPs extension

A.1 Main Functions

- **setup_scene**: Sets up the scene by adding a ground plane, physics context, Xform references for shuttles, flyways, trays, and robots.
- **setup_post_load**: Performs additional setup after loading the scene, including adding lab setup, shuttles grid, flyways, and setting up the ROS action graph.
- **get_controller_data**: Reads a YAML file from MoveIt config and parses the information about each robot. For each planning group, it constructs a dictionary entry with data as MoveIt *planning_group* name, *joint* names or *eef_link* name.
- **read_instructions_from_yaml**: This method reads the recipe instructions from a YAML file. It returns a list of instructions read from the file.
- **move_to_joint_state**: Moves a robot to the desired joint positions using the ROS planning group and joint state.
- **move_to_pose**: Moves a robot to the desired pose (position and orientation) using the ROS planning group, position, and orientation.
- **move_along_cartesian_path**: Maneuvers a selected robot along a predetermined path, provided by the waypoints parameter.
- **move_shuttle_to_target**: Sets the target position for a shuttle robot identified by *xbot_id* to the specified *target_x* and *target_y* coordinates.
- **attach_object**: Attaches a specified item to the shuttle, it does this by adding a physics callback that executes on each simulation step.
- **on_sim_attach_object**: Callback function that gets executed at each simulation step. It sets the position of the item to be the same as the shuttle's position, essentially making the item appear attached to the shuttle.
- **gripper_control**: Controls the gripper of a robot identified by the planning group to either open or close.

- **get_eef_link_position**: Retrieves the current position of robot arm's end effector link by accessing the transform (translate) attribute.
- **get_shuttle_position**: Provides the current position of a shuttle ,identified by its ID, by accessing the transform (translate) attribute.
- **get_gripper_joints_position**: Gets the current joint positions of a gripper associated with a robot hand.
- **get_joints_position**: Retrieves the current positions of the joints of a specified robot arm.
- **has_reached_position**: Verifies whether a robot arm, its gripper or a shuttle has arrived at a desired position within a specified tolerance range.
- **print_action_times_summary**: Prints a summary of times for each action performed by the robot, including the total time and the number of times the action was completed.
- **execute_actions**: Carries out a sequence of actions defined in the recipe. If the preceding action has been successfully completed, it retrieves the next action from the sequence and performs it based on its type.
- **on_start_experiment_event_async**: Asynchronous function to start an experiment from the GUI.
- **on_impulse_event**: Enables the impulse event of the action graph for each physics step.
- **on_sim_step_check**: Callback function that checks whether the robot, gripper or shuttle (planning group) has reached the desired position.
- **on_automatic_execution**: Callback function to automatically execute actions from the recipe.
- **setup_pre_reset**: Asynchronous function executed before resetting the simulation setup.
- **setup_post_reset**: Asynchronous function executed after resetting the simulation setup.
- **world_cleanup**: Cleans up the world after the simulation.
- **sim_xbots_movement**: Moves simulated xbot robots in the simulation towards their target positions. No collision avoidance implemented.
- **sim_xbots_movement_collision**: Moves simulated xbot robots in the simulation towards their target positions while considering collisions with other shuttles.
- **read_xbots_positions**: Reads the positions and orientations of shuttles from the physical setup and updates them in the simulation.

- **send_xbots_positions**: Sends commands to move the shuttles to unique targets in the physical setup only if the xbots are idle and a certain time has passed since the last update.
- **create_random_coordinates**: Generates random coordinates for each shuttle in the environment, ensuring uniqueness.
- **sample_motions**: Performs a sequence of linear motions for a given xbot ID in the physical setup, specifying target positions, maximum speed, and maximum acceleration.
- **wait_for_xbot_done**: Waits for an xbot with the specified ID to finish its current action in the physical setup before continuing.
- **connect_pmc**: Connects to the PMC, gains mastership, and activates the xbots.
- **create_pose_msg**: Creates a Pose or PoseStamped message object based on the given position and orientation.
- **platform_pos_to_coordinates**: Converts Acopos platform positions to platform coordinates, with an option to apply MoveIt frame offset for the robot arms.
- **sim_xbots_movement_bfs**: Moves simulated xbot robots in the simulation towards their target positions using the breadth-first search (BFS) algorithm to avoid obstacles.
- **bfs**: Performs the breadth-first search (BFS) algorithm on a grid, finding the shortest path between a start position and an end position.

Appendix B

How to run the experiments

B.1 Running the experiment in Nvidia Isaac Sim

Steps to conduct the experiments presented in section 5.1:

- Open a terminal and run *roscore*.
- Launch Nvidia Isaac Sim and select the *omni.isaac.ros_bridge* extension in the window before starting.
- Load the MAPs extension, press *LOAD* and finally, start the simulation
- In another terminal, run *roslaunch isaac_moveit kuka_isaac_execution.launch*

Once these steps are done, the experiments can begin.

B.2 ROS connection to Kuka

Steps to connect to the Kuka robots using ROS and Kuka Var Proxy solution:

1. Set the robot IP to 192.168.1.15 and the port to 7000
2. Uploade and run the Kuka Var Proxy program on the Windows 10 side of the pendant.
3. In the ROS launch files, set the target IP address and port number, which in this case is the robot IP and port numbers, 192.168.1.15 and 7000, respectively.
4. On the teach pendant, select and run the program that sends the robot to an arbitrary home position and after it starts a loop in which the robot now can be moved from the ROS side. The program has to be run in T1 (teaching mode where the velocities are limited) until the robot enters the loop and only after the loop entry, the robot can be set to AUT (Automatic mode with unlocked velocity).
5. Once the robot is in AUT, the ROS side can be started from the Ubuntu terminal by running the following command: *roslaunch kuka_moveit_config demo.launch*. This command will launch the hardware interface that connects to the robot and the Rviz software from which the robot can be moved.

By default, the *demo.launch* file will connect to the 192.168.1.15 robot IP, but there is the possibility to chose different robots by giving the *Robot_IP:=192.168.1.xx* argument. The xx part of the IP address can take values from 11 until 15, 11 representing Kuka 1 and 15 representing Kuka 5.

B.3 Acopos 6D connection from MAPs environment.

The steps to follow to drive the shuttles on the platform are as follows:

1. Set local computer IP in range with PMC IP to establish TCP/IP connection.
2. Using the Isaac Sim extension GUI, press the *connect PMC* button to connect with the PMC and gain mastership.
3. Press *Start Real Setup* to enable sending and reading the position and orientation of the shuttles in the setup.
4. Run the experiment by pressing *Start Experiment* button to start moving the shuttles when needed.