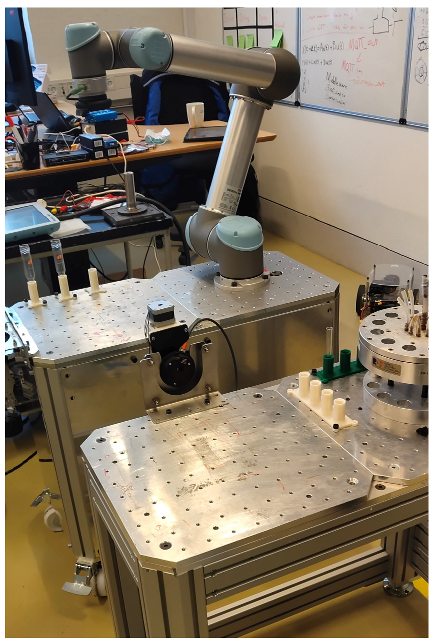
Automatic Synthesis of Molecules by Robot

-Using UR5 Robot and RG2 Gripper-

Project Report

VT4 - Haoxiang Shi



Aalborg University
Department of Materials and Production
Fibigerstræde 16
DK-9220 Aalborg

Manufacturing Technology
Department of Materials and Production
Fibigerstræde 16
DK-9220 Aalborg
www.mp.aau.dk/

Title:
VT4-Project

Theme:
Automatic synthesis
Robot controlling
Synthesis of Molecules

Project Period:
From February the 1st 2022
To June the 2nd 2022

Project Group:
VT4-Haoxiang Shi

Participant(s):
Haoxiang Shi

Supervisor(s):
Ole Madsen

Copies: 1

Page Numbers: 61

Date of Completion:
May 31, 2022

Abstract:

Chemists hope to liberate themselves from tedious, repetitive experiments and devote themselves to more creative work. Therefore, this project will try to introduce cooperative robots to replace manual operations. After visiting the chemical laboratory of AAU, this report will analyse and deconstruct the task and the dilution experiment of the quinine solution involved. On this basis, the concept of decoupling complex operations into several basic robot actions is proposed. The corresponding relationship between manual operations and robot actions is shown in one flow chart. In the practice part, this project establishes a set of robot workbench that meets the requirements of the chemical laboratory and is also consistent with reality by measuring the size and position of the existing equipment in the laboratory in the simulation software. Write the robot action sub instructions, try to combine them into the main program arbitrarily, and finally complete a set of user-defined experimental operations. Although there are still problems in the control of syringes, the rest have been tested and verified to meet the project requirements.

Contents

Preface

This project was made by 4th semester Master student, VT4-Haoxiang Shi, from Manufacturing Technology at Aalborg University from the 1st of February to the 2nd of June 2021. This project was supervised by Ole Madsen (Department of Materials and Production) and Casper Steinmann(Department of Chemistry).

Acronyms

- AAU : Aalborg University
- TCP : Tool Center Point

Signatures

Haoxiang Shi

Haoxiang Shi

Signature

Part I

Introduction

Chapter 1

Introduction

This chapter introduces the project background of the chemical experiment from automation to intelligence and describes the advantages and significance of using robots to replace traditional manuals.

1.1 Project background

With the development of information technology, numerical control and automation technology have been widely used in many fields. Compared with manual manufacturing, the advantages of regularity, high efficiency and controllability of robots are undeniable after the improvement of the 21st century[1] . However, in chemical manufacturing and especially in the research of new drugs, manual, repetitive operation still accounts for most of the proportion. Chemists are eager to break away from inefficient, traditional operations and have more time to exert their imagination, experience and theoretical knowledge and invest in more innovative research[2] . Therefore, since the 1990s, experts from different disciplines have been working together to study efficient and automated molecule synthesis laboratories [3]. Fortunately, significant progress has been made in some areas. For example, many chemical laboratories have put forward many concepts of rapid preparation of drugs and tried to build them. The fundamental goal of accelerating the experiment has achieved phased success.

At the current time node, how to make the chemical laboratory more intelligent replace the high efficiency of synthesis has become the focus of a new round. Pharmaceutical chemists need efficient synthesis tools that can operate freely according to the analysis results, rather than expensive production machines that can only perform a single type of experiment[4]. Ideally, the intelligent laboratory should have the following characteristics: (1) It can perform various experimental operations; (2) The sequence of operations and the use of molecules should be entirely customised; (3) The process should be simple and conforms to the intuition of non-robot professionals; (4) The interface should be intuitive and concise.

This report will try to apply robots to the chemical laboratory, looking for the possibility of replacing manual synthetic to reduce the long-term impact of the environment on the health of operators and avoid the risks brought by some human factors. Under this assumption, non-robot professional experimenters can also randomly combine a set of main programs that will perform complete chemical experiments according to the existing subprograms in the software (or even only keywords). The customised main program will be directly uploaded to the robot, and the robot can experiment according to the expectation of chemical analysis. At present, the project is only in its early stage, and there is still a lot of future work.

Part II

Problem Analysis

Chapter 2

Context Analysis

This chapter will introduce AAU's chemical laboratory, reference experiments, and tools. Through the decoupling of chemical operations, it can be converted into the basic actions that the robot arm can perform.

2.1 Task analysis

This section will focus on the specific content of the task and sort out the work flow of this project.

Briefly summarise the mission background and objectives in section 1.1. Because the operations of chemical experiments have strict norms, there has been no apparent change in the past century. Chemical practitioners hope to realise the iteration of operations through automation technology. The characteristics of the robot, such as safety, stability and fast speed, make it possible for the robot arm to replace most of the repetitive operations in chemical experiments. Therefore, this project hopes to use robots to complete some traditional basic human operations. Ideally, a robot can perform automatic synthesis of the molecules. The conceptual diagram is shown in Figure 2.1

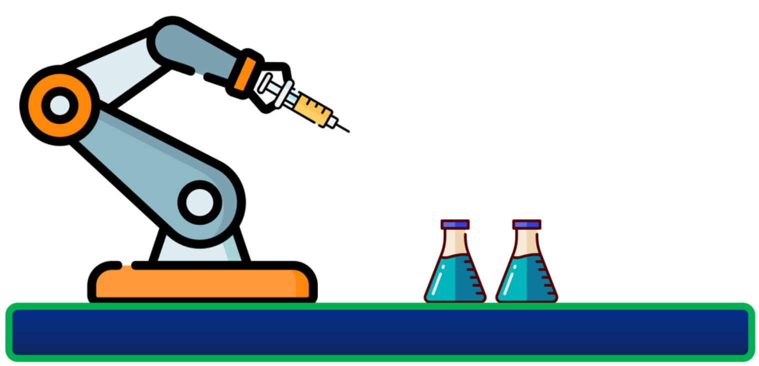


Figure 2.1: The Conceptual diagram of project task[5][6][7]

The execution program of the robot is based on the combination of several basic actions, so one of the critical steps is that the chemical experiment operations involved should correspond to the robot's actions. In other words, the experimental operation needs to be decoupled into basic actions with high repetition, and the robot will reproduce these sub-operations. The entire robot experiment will be realised at the program level by establishing several basic action program modules and reorganising them, including movement and grabbing. Then combine the subprograms into the main program according to the sequence. The chemist only needs to click the execute bottom of the main program to order the robot complete the experiment. Summarise the above process into the following Figure 2.2 and this project will complete subsequent project on this basis.

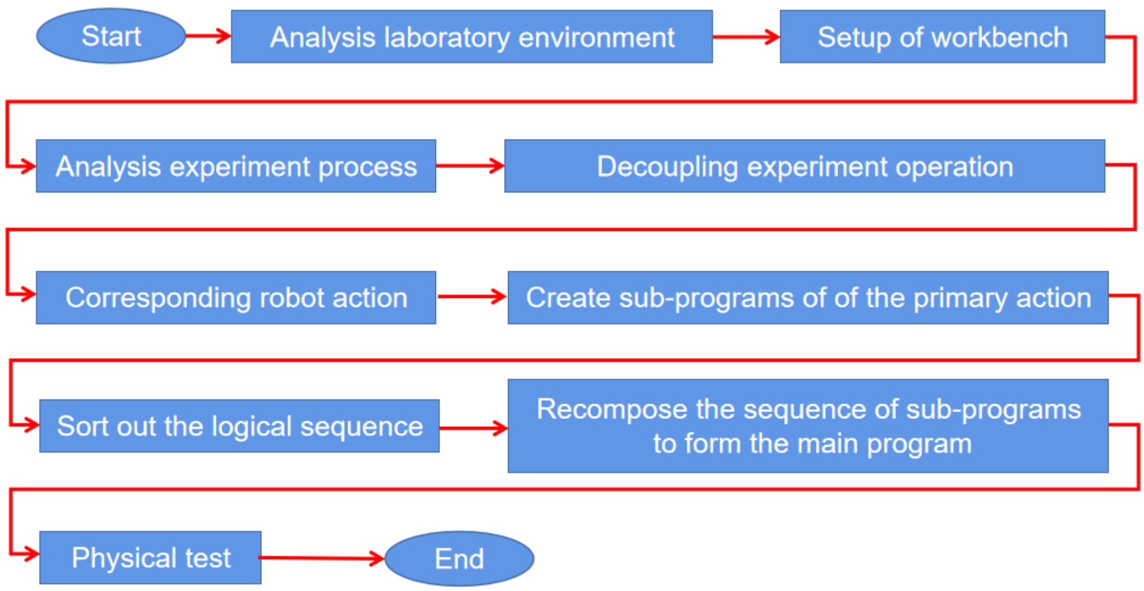


Figure 2.2: The work flow of the whole task

2.2 Analysis of chemical part

This section will focus on chemistry-related parts, including chemical laboratories and chemical experiments. The human operation is simplified and classified into different robot actions. Finally, according to the task requirements, the logical structure of robot action will be summarised.

2.2.1 Reference chemical experiment

Taking the dilution of quinine solution as the example, this project attempts to simulate the basic chemical operations and establish a physical model which is still easily operable in the real environment.

The whole human operations except the calculation of diluting quinine solution are described as follows. The specific experiment video demo can be seen in GitHub:Automatic-synthesis-of-Molecules-by-UR5:

- Step 1: prepare tools, undiluted quinine solution and distilled water. The tools include a beaker or test tube containing the compound and a syringe that can quantitatively absorb the liquid (the measurement of the molecules needs to be calculated and weighed in advance).
- Step 2: the syringe is moved directly above the quinine solution container, and the quantitative quinine solution is extracted from the container in batches and loaded into different empty containers in turn.
- Step 3: clean the syringe. The specific operation is to align the syringe with the clean water tank, discharge and extrude the residual liquid and air in the syringe, stretch into the clean water tank, suck enough clean water, lift the syringe and release the internal liquid. This process needs to be repeated 3-5 times to ensure no residual quinine solution in the syringe.
- Step 4: the syringe is moved directly above the distilled water container, inhale the required amount of distilled water in batches, and move them to the required containers.
- Step 5: clean the syringe (the operation is precisely the same as in Step 3).

2.2.2 Convert to robot's action

According to the chemical experiment operation introduced in the subsection 2.2.1, the specific operations can be sorted to facilitate the subsequent transformation of specific human actions into robot actions.

Because the project requires the implementation of the action execution of the robot, the classification standard of chemical operations must also refer to the types of basic actions of the robot. According to the different execution parts, it can be divided into the action performed by the robot and the tools. Tools can also be divided into gripper and syringe. Therefore, all operations moving to a certain position belong to the robot's path planning, including moving to the top of the test tube and moving to the top of the pool. Picking up and placing the test tube correspond to the opening and closing of the gripper. All liquid transfer belongs to the scope of syringe work. The **??** shows the above action categories.

	Robot	Gripper	Syringe
Chemical operations	Movement of tube or solution	Grab the test tube	Liquid transfer
Robot's actions	Path planning of robot	Gripper control	Syringe control

Table 2.1: Classification of both human and robot actions

The sorted actions need to meet the following requirements so that the actions can be subdivided again: first, clarify the specific coordinates of different containers and tools; Second, the end actuator of the robot must accurately move to the inside or right above different containers (there will be no collision in this process); Third, the syringe needs to be controlled as planned; Fourth, the gripper needs to be controlled as planned; Fifth, the action path of the robot arm must be freely planned, and non-robot professional operators can quickly set or modify the ideal action path.

Robot's action	Action requirements	Critical measures
Path planning	Obtain the objects' position	Establish an unified coordinate system
	Obtain the objects' shapes	Measure the objects' dimensions
	Design the robot's movement route	Set way-points and path types
Gripper control	Open or close	Create a control program
Syringe control	Push or pull the syringe handle	Create a control program

Table 2.2: Action requirements and corresponding concrete measure

Referring to the behaviour requirements in Table 2.2, it is necessary to refine the actions into basic actions and set completion goals. The premise of determining the position coordinates is to establish a complete setup and a unified coordinate system with the robot as the origin. All containers and tools will occupy specific coordinates and attitudes in the reference coordinate system of the robot. Secondly, the fixed-point coordinates above the test tubes are determined according to the length of the tool as the preparation point, which is convenient for the robot to extend more smoothly into the tube. Third, fix the syringe and the handle, stretch or push the syringe handle through the transmission of the motor and screw rod to control the syringe. Fourth, the program directly controls the opening and closing of the gripper. Fifth, the above operations will exist in the different basic programs, and the operations can be completed by directly calling these programs to form a main program when necessary.

Based on the dilution process of quinine solution in subsection 2.2.1 and the corresponding relationship between human operations and robot actions in subsection 2.2.2, the logic structure shown in the figure below can be summarised.

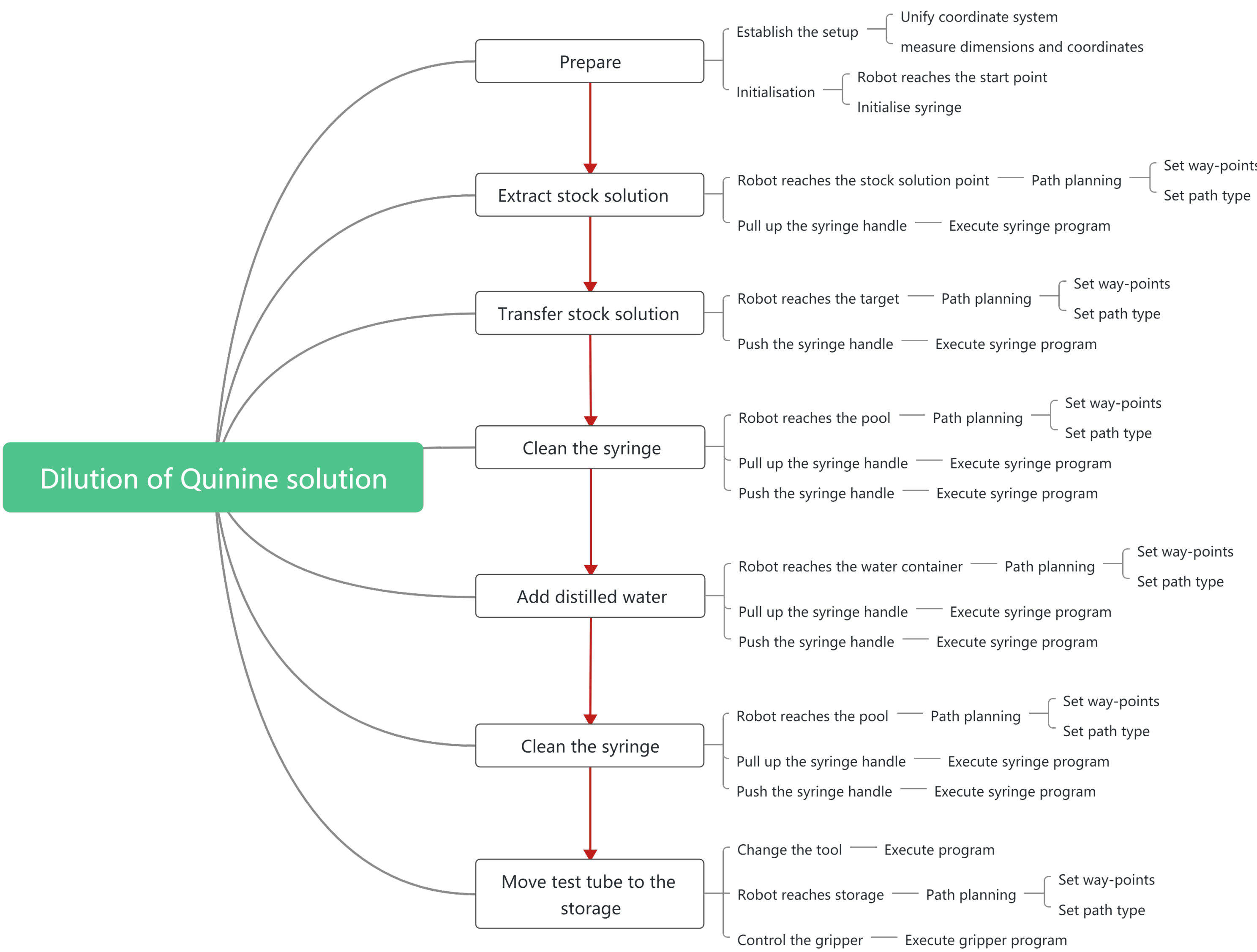


Figure 2.3: The chemical operation flow of Quinine solution dilution and correspondence between operation and robot action

2.2.3 AAU chemical laboratory

Since this project is based on the chemical laboratory of AAU, this section will introduce the chemical laboratory and chemical tools in detail and summarise the expected robot workspace.

Figure 2.4 shows one worktable in the AAU chemical laboratory, which is called a fume cupboard. The lab mainly takes several of these closed worktables as the operation platform. The specific size of the platform is **2m × 1.2m**, and the operator needs to open the glass window and carry out the experiment on the platform. Therefore, the standard workbench will be placed with conventional chemical tools such as a heater, test tube and weighter.

Figure 2.4: The standard work station in the AAU chemical laboratory

The syringe and beaker shown in Figure 2.5 are usually used in the experiments. The syringe can adjust the volume of liquid inhaled through the knob at the handle end and display it on the side. Once the suction volume is determined, each push and pull of the syringe handle will only draw a fixed volume of liquid. Beaker is a standard container for liquid, which will be replaced by test tube in the project.

In addition, a heater and a distilled water pool, as common tools in chemical experiments, should also be added to the project. Refer to Figure 2.6 for the shape of the heater. The 12 groups of inner holes of the heater are used to place the test tube to be heated. Unique inner holes lead to the shape of the test tube that can be heated is also unique. There are no restrictions on the the pool's shape. More figures of laboratory can be accessed in GitHub:Automatic-synthesis-of-Molecules-by-UR5

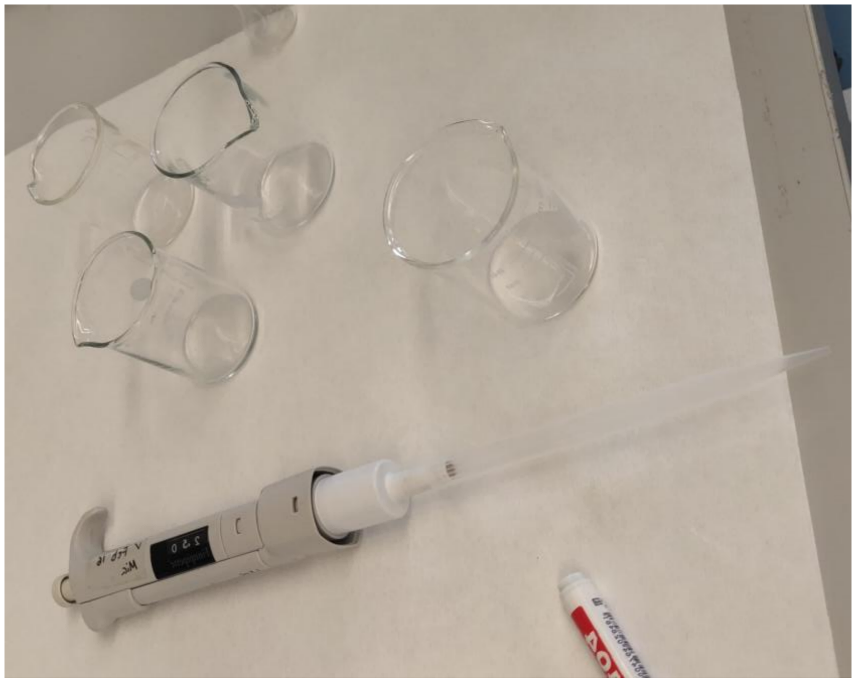


Figure 2.5: The syringe and beaker in the AAU chemical laboratory

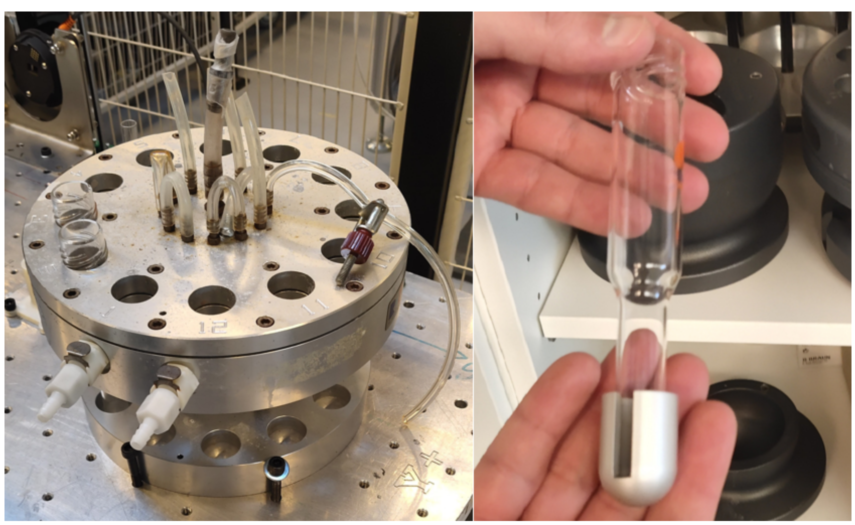


Figure 2.6: The heater and teat tube in the AAU chemical laboratory

2.3 Technical analysis

This section will specifically introduce the robot type and tools used in this project, and show their appearance, advantages and disadvantages.

2.3.1 Robot

This project adopts the UR5 cooperative robot arm of the universal robot prepared in the AAU laboratory, as shown in Figure 2.7 [8]. This type of robot has the following advantages[9]:

- The max payload of UR5 robot is 5kg
- The 6-axis robot arm has 6 degrees of freedom (DoF)
- The working radius is 850mm
- Graphical and 3D visual programming, easy to learn and use
- High precision, which can reach 0.02mm

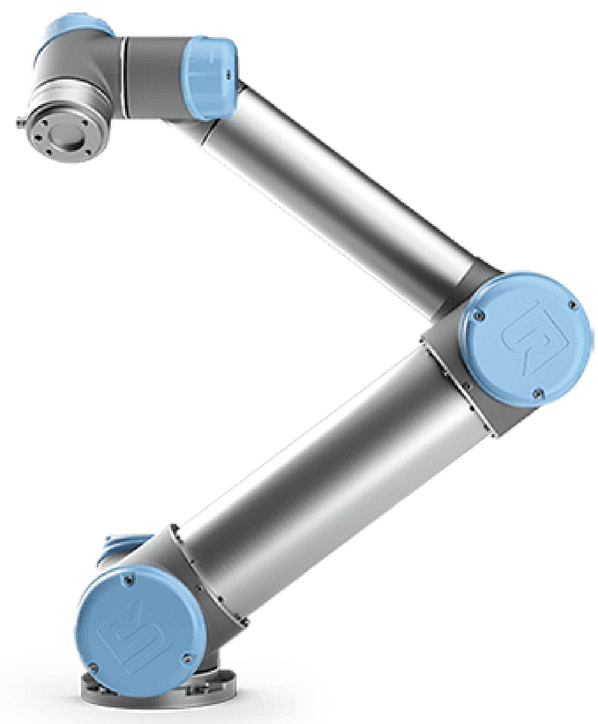


Figure 2.7: The UR5 robot[8]

2.3.2 Gripper

The RG2 gripper[10] of OnRobot is used in this project. Please refer to Figure 2.8. RG2 is a flexible two filter robot gripper, which has the following advantages[11]:

- The maximum stroke is 110mm, which is enough for the requirement of this experiment;
- The finger end can be customised and replaced, and containers of different sizes can be used more flexibly;
- The maximum payload of the gripper is 2kg;
- Automatically calculate the payload in use to reduce the complexity of programming;
- Automatic depth compensation;
- Automatically calculate tool center point (TCP)

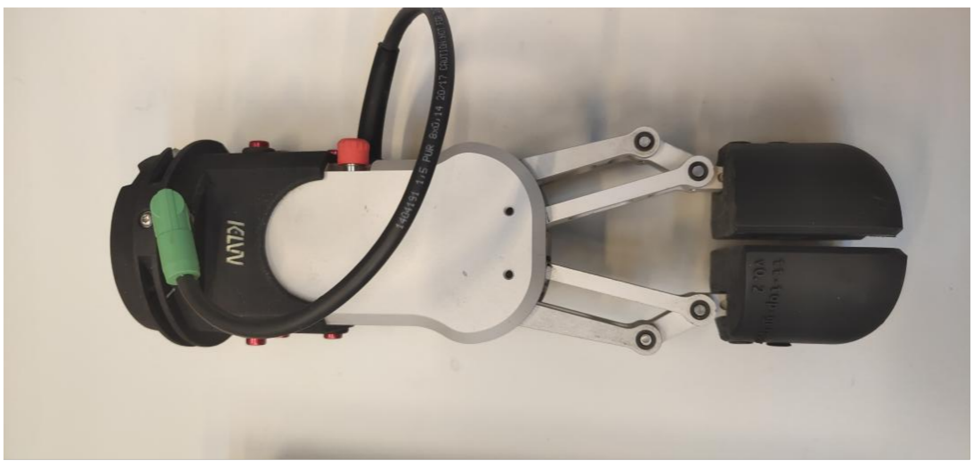


Figure 2.8: The RG2 gripper used in this project

2.3.3 Syringe

First of all, the syringes used in the chemical laboratory are different from that used in this project, but the principle is almost the same. For the shape and use method of syringes in the chemical laboratory, please refer to subsection 2.2.3.

As shown in Figure 2.9, the syringes used in the project are entirely developed by AAU. The Arduino board (esp32) is used for data processing and control. The command sent by the computer is received through the communication protocol to control the forward and reverse rotation of the motor. The rotation of the motor will drive the screw rod to convert the rotary motion to up and down linear motion. The fixing part of the screw rod is connected with the tail of the syringe. When the motor rotates forward, the screw rod will drive the syringe handle to move up, while the syringe body does not participate in the movement, so the stretching movement of the syringe can be realised. The pushing movement of the syringe can be completed similarly. The motion control system used in the syringe is open-loop control so that it can be further optimised in the future.

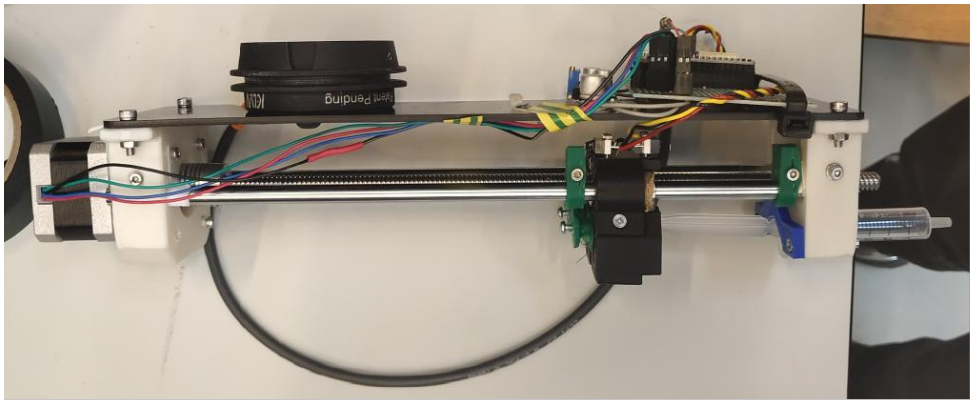


Figure 2.9: The syringe tool used in this project

2.4 Conclusion

The chemical laboratory expects to place one or more cooperative robots in an enclosed space of 2m $\times$ 1.2m. The robot can automatically carry out the chemical operation according to the quickly written main program. The object's position can be planned without obtaining the position coordinates through camera and machine vision processing. Meanwhile, all molecules needed have been placed in the predetermined position before the robot operation begins. This project is based on a simple solution dilution operation and does not involve more complex actions. Finally, robots and tools are readily available in the AAU laboratory and do not need to be manufactured separately.

Chapter 3

Problem Analysis Conclusion

Based on the analysis of the second chapter, this chapter will extract the relevant requirements of the project experiment and the standards to verify whether the project meets the requirements.

The final problem formulation can be stated: *A non-robot expert can make the robot perform customized chemical experiments through simple operations.*

3.1 Requirements

The experiment of this project must meet the following requirements to ensure that the final experimental effect meets the expectation:

- The sizes of the objects in the simulation environment should be consistent with those in the real environment, which means the robot will not collide with objects in moving;
- The coordinates of objects in the simulation environment should be consistent with those in the real environment. For example, the robot can accurately move to the inside of the test tube;
- The workbench layout in the simulation environment should conform to the operation intuition. For example, the preparation and completion area of the compound should be close to the window, etc. Refer to section 4.2 for specific requirements;
- The control program of the tool shall be integrated into the simulation software;
- The path planning of the robot should be intuitive and straightforward;
- The subroutine should be prepared in advance. The chemical operator only needs to select the required subroutine to establish a set of the main program to complete a task;
- The simulation results must be verified in the physical environment;
- Accidents such as collisions shall not occur during the movement of the robot;
- The action of the robot needs to increase the speed control to ensure that the test tube will not be damaged in the process;
- Ensure the integrity of the experimental instruments during the physical experiment;
- When the robot has multiple subroutines running, it should be ensured that there is no confusion between these programs;

- This project will focus on the reference experiment, the dilution of Quinine solution;
- A non-robot expert should be able to use the system.

3.2 Test standard

From the conceptual simulation to the physical practice of the project, researchers need to have a complete set of test standards to ensure that the project's effect meets the expectation. The following operational test criteria will satisfy this requirement:

- The tool coordinate system must be consistent with the actual situation, including parameters such as angle, coordinate and robot attitude;
- The home of the robot should be consistent with the actual situation. That is, when the program of returning the robot to the 'home' position is sent to UR5, the posture of the robot should be consistent with that in the simulation;
- The robot coordinate system is consistent with the actual situation. That is, when the robot receives the movement to some specific points, it will make an action utterly consistent with the simulation effect;
- The size and position of the objects must be consistent with reality. That is, when the robot receives the program moving to the interior of the object, it will make an action completely compatible with the simulation effect;
- When the robot receives the control program of the tool, the tool will run as expected;

Part III

Simulation

Chapter 4

Concept Design

This chapter will design two simulation concepts according to the analysis conclusions and requirements. Then try to run the test program in the simulation environment to compare the advantages and disadvantages of the two concepts. Finally, a more appropriate concept will be selected for subsequent actual development and commissioning.

4.1 Simulation software

In this project, RoboDK will be used as the simulation software for path planning. RoboDK is a simulation program[12] specially used for various robot applications. Through three-dimensional modelling, a simulation environment with a scale of 1:1 to restore the actual scene can quickly establish an intuitive and straightforward robot coordinate system, world coordinate system and other common reference systems. Coordinate data can be transformed between these frames. Based on exact coordinates, quickly setting the target at the desired location is available. RoboDK can quickly create a complete program according to the required point positions and their sequence, that is, the 'path planning' of the robot. In addition, RoboDK also supports offline self-editing Python programs, which can be run directly. This helps to implement some infrequently used customisation functions. The following will summarise the advantages of the software[13]:

- The interface is intuitive, and even non-robot professional operators can get started very quickly;
- It has a vast library, which can be called directly, such as UR5 in this experiment;
- Directly upload self-made 3D modelling;
- The motion path of the robot can be easily planned;
- The planned path can be easy re-edited to present completely different tasks;
- The software has a variety of standard post processors, which can be directly connected to the mainstream robot. In this way, the software can automatically generate the program and send it to the robot;
- Have offline python programming function;
- It can automatically optimise the path of the robot to avoid singularity, axis restriction and collision;

The interface is shown in Figure 4.1 below:

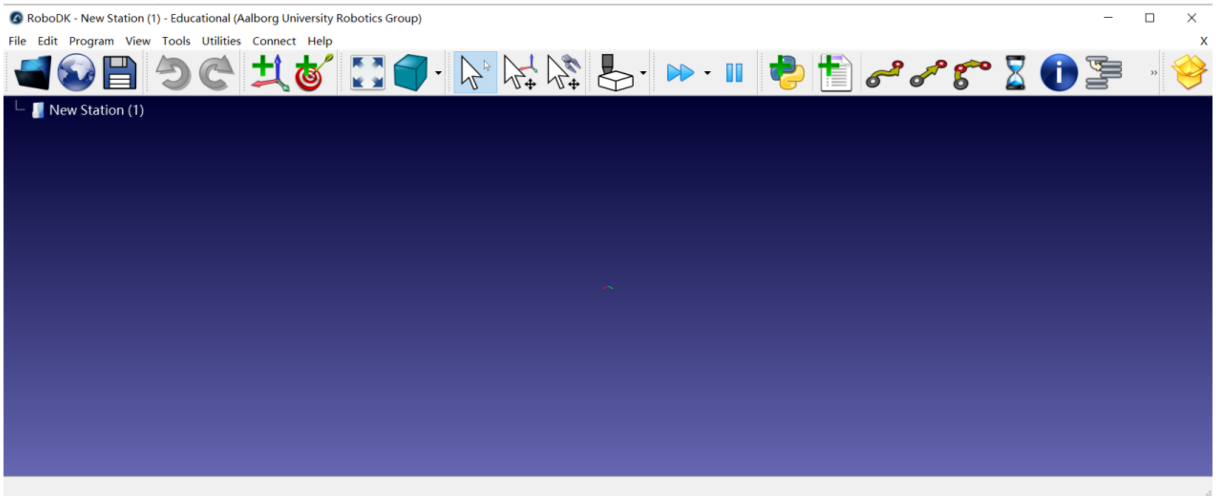


Figure 4.1: The interface of RoboDK

4.2 Setup requirements

In order to make it easier for operators to observe and control the experimental process and even stop the robot in time when there are errors in the experiment, the layout of the workbench must meet the following zoning requirements:

- Ensure that the working space of the robot is enough to cover all objects on the table;
- Place objects with different functions in different areas;
- The desktop should be 2m to 1.8m long and 1m to 1.2m wide.
- It is better to leave enough space between zones to prevent an unnecessary collision when the robot moves;
- The operator should place the compound prepared in advance nearby to prevent accidents when placing the compound;
- The operator should collect the configured solution nearby to prevent accidents;
- The heating and clean pool can be placed away from the window;

4.3 Concept with one robot

In view of the above requirements, compared with the workbench layout in the chemical laboratory, continue the setup already arranged in the laboratory for expansion, that is, use a single mechanical arm for operation and partition the desktop more carefully. The conceptual simulation diagram of the concept is shown in Figure 4.2 below. The desktop size is 2m $\times$ 1.2m, meeting the experimental requirements. Note: since it is only a conceptual diagram, the dimensions of the objects and tools used in the drawing are not completely consistent with the physical objects and can only be similar. However, in the practical operation stage, the object will be modelled again according to the size to ensure that the simulation results are consistent with reality.

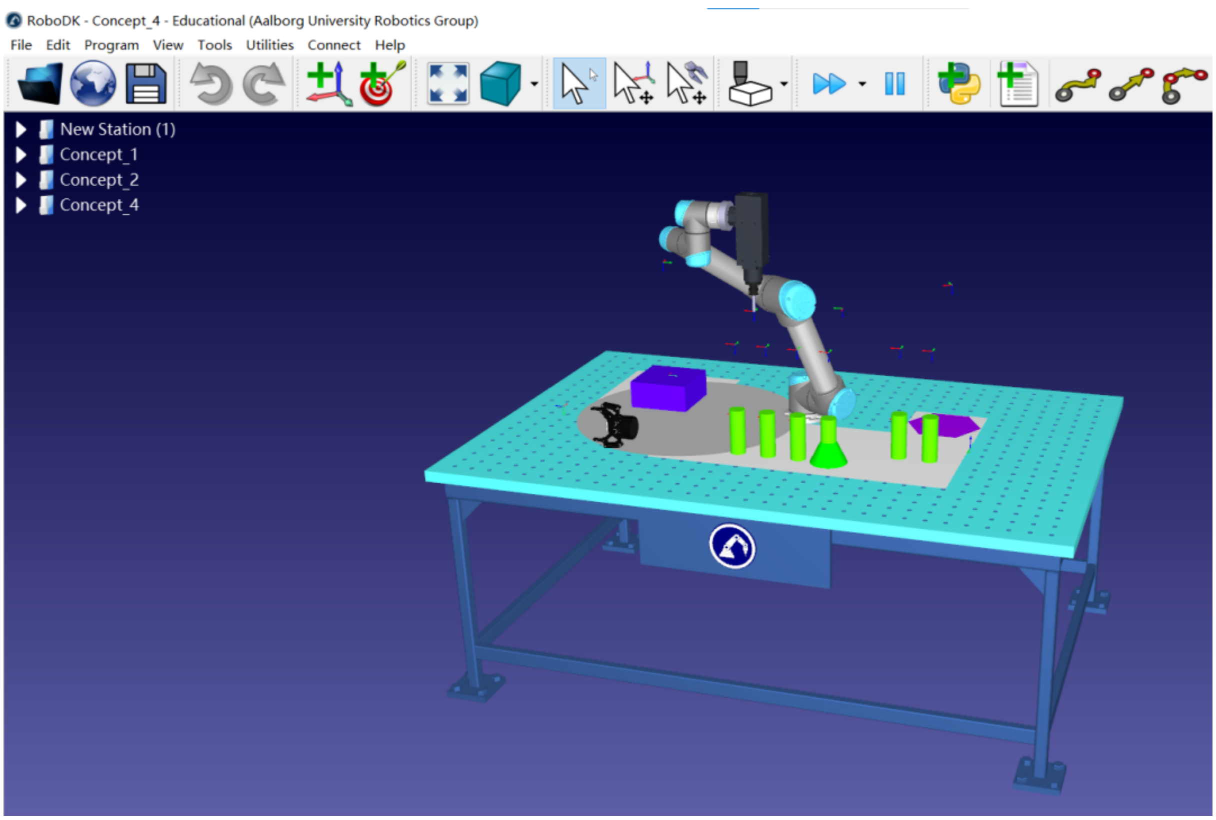


Figure 4.2: The simulation for concept 1

First, areas of different colours can be distinguished from the top view of Figure 4.3. These areas also represent different functions. According to the colour division, the functions can be expressed as follows:

- Yellow area: preparation area. The operator puts the compounds and containers required for the experiment in this area, and the robot arm automatically and sequentially looks for these items needed according to the plan. This area is close to the window, which is convenient for the fixed-point placement of the operator;

- Purple area: experimental area. The compound was transferred from the yellow area to the pink area for the experiment. The specific practical operation of the robot will be completed in this area as far as possible. Since this area is close to the window, the operator can more conveniently observe the progress of the experiment and judge whether there are problems in the current operation. If there is a problem, the process of the robot should be stopped immediately to stop the experiment;
- Green area: finished product placement area. The prepared compound will be clamped through the robot arm and placed in the area. This area is also close to the window, which is convenient for the operator to take;
- Blue area: distilled water holding area. A small pool will be arranged in this area. In order to prevent the solution from remaining in the syringe during the experiment, it is necessary to clean the syringe. After the syringe is transferred to the area, repeat the two basic actions of sucking liquid and releasing liquid five times;
- Red area: heating area. A heater will be placed in this area. When the experimental compound needs to be heated, the mechanical arm will clamp the test tube containing the solution and transfer it to the heater;

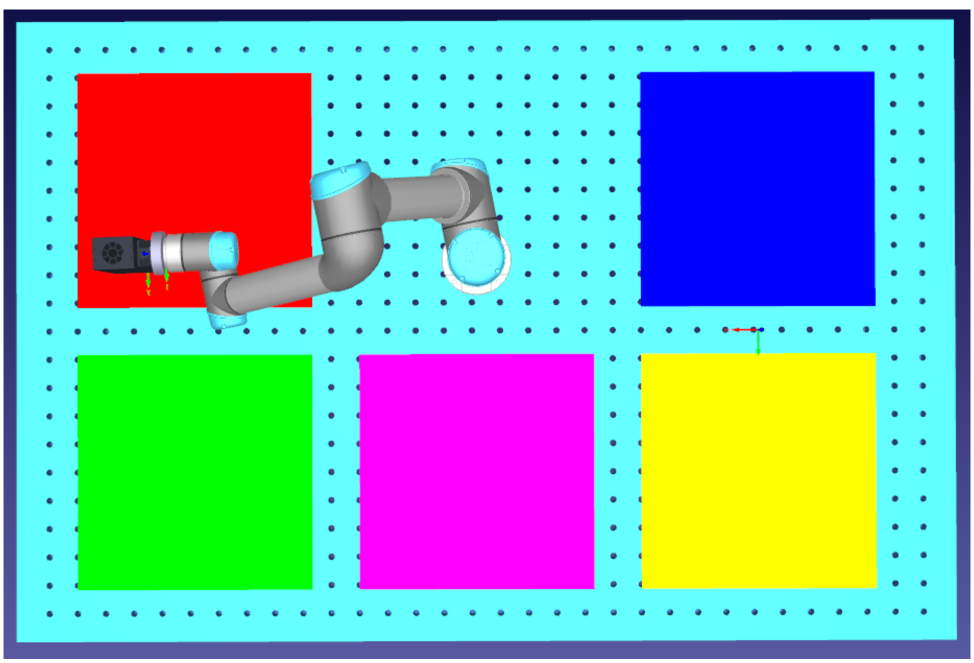


Figure 4.3: Areas of different colours have different functions

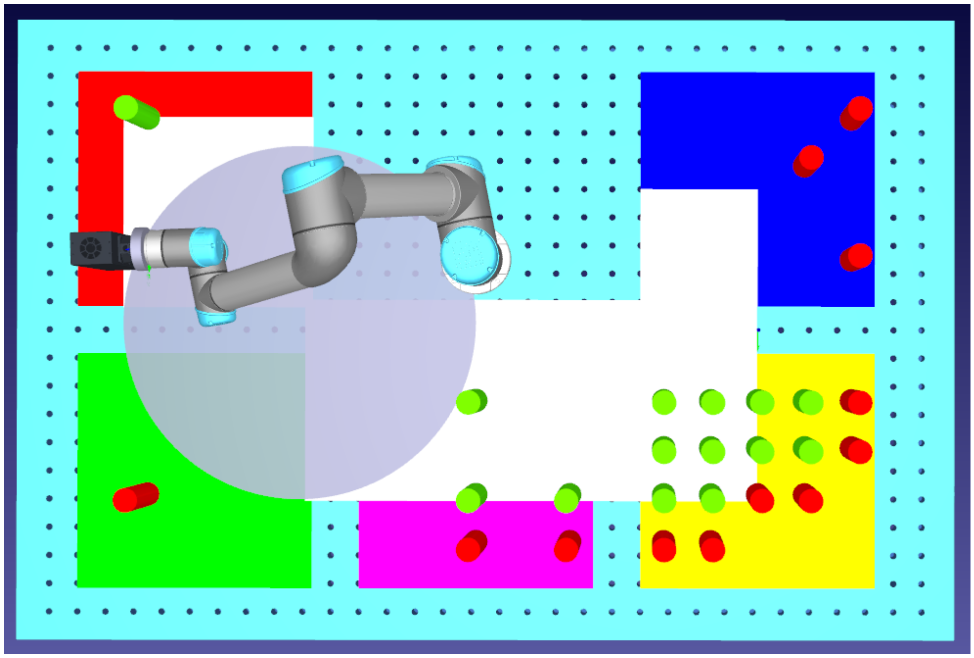


Figure 4.4: The available working area for concept 1

Secondly, two cylinders of different colours can be distinguished from the top view of Figure 4.4. Among them, the red cylinders and their areas outside are not in the workspace of UR5. In other words, the robot in the areas outside the red cannot reach and make any operation. The working space within UR5 is green. Note that although the working range of the end flange can be directly shown in RoboDK, as shown in Figure 4.5. However, because the tools carried at the end need some additional space, the working range is different from the actual 'working space'. It is concluded that the workspace of the end flange is larger than the workspace that can be reached by the end of the tool, which is obtained after many attempts in the simulation environment.

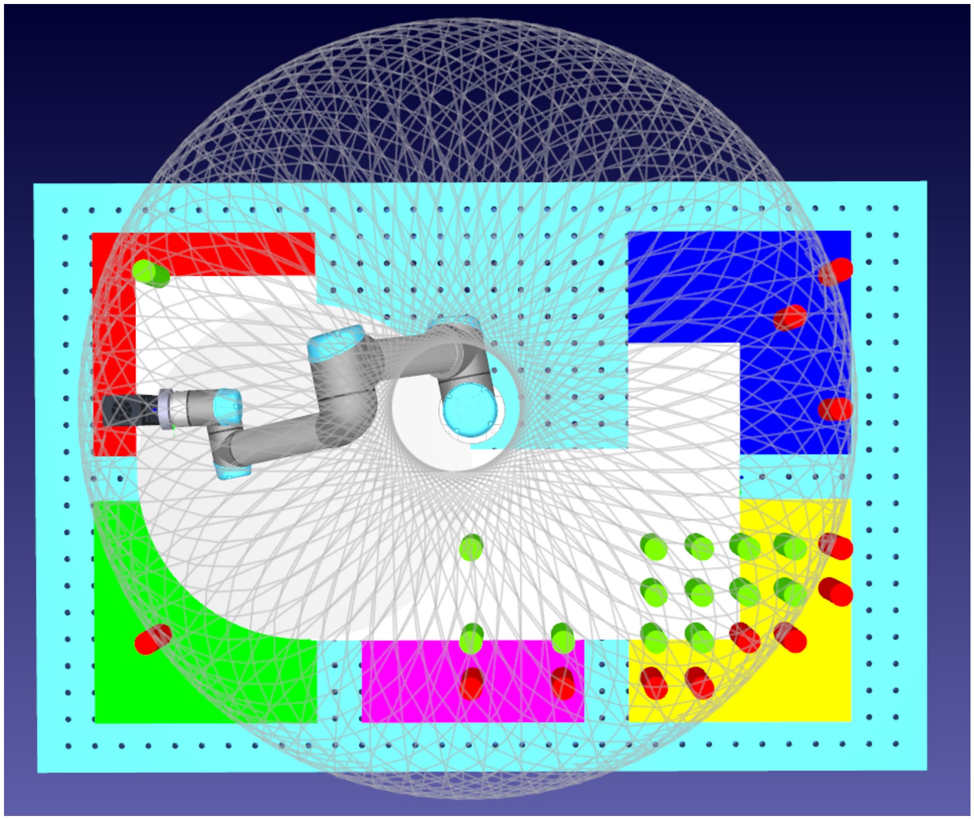


Figure 4.5: The work space that the UR5 can reach

To sum up, the concept has the following disadvantages:

1. The working range of the UR5 robot is obviously smaller than the usable desktop size, resulting in a waste of area and a reduction in the number of equipment arranged, which may not be suitable for chemical experiments with high complexity;
2. A single robot experiment must involve tool replacement. Because the two most basic operations of the experiment are to grab and move the liquid, it is doomed that two tools should be used to complete it, so a single tool can not undertake even the most straightforward experiment. On this basis, considering how to replace tools, placement of tools, etc is needed.

The advantages of this concept are:

1. Low cost. The price of a mechanical arm is between thousands and tens of thousands of Danish Kroner. Each additional robot arm will have a significant impact on the cost;
2. The initialisation of robot is simple. Each additional robot needs an extra series of operations such as initialisation, calibration, etc.

The advantages and disadvantages are sorted in the following Table 4.1:

Advantages	Low cost	Simple initialization
Disadvantages	Small working range	Tool replacement

Table 4.1: The advantages and disadvantages of a single robot concept

In the conceptual simulation, the simulation program is compiled in this project, which can completely simulate the dilution experiment of quinine solution. The final analogue video effect can be downloaded and viewed in GitHub:Automatic-synthesis-of-Molecules-by-UR5.

4.4 Concept with two robot

Two UR5 are arranged relatively, and only one kind of tool is installed on the end flange of each robot. This concept requires the same concept for desktop partition, and it should be classified and sorted as much as possible to make it easy to operate. Note that the dimensions of the objects used in the concept map are consistent with those in concept 1. New modelling will be used for path planning to ensure the consistency of dimensions. The specific conceptual simulation diagram is shown in Figure 4.6 below, and the desktop size is also 2m $\times$ 1.2m.

Referring to the Figure 4.6, the prepared compounds need to be placed in the purple area. Taking this project as an example, the area where quinine solution, stock solution and distilled water are placed; The green area is the space for placing the finished compounds; The orange area is the operation area, and the reaction mainly occurs in this area. Red is the heating area used to place the heater; blue is the water tank, which is used to clean the residual liquid in the syringe. all of the functions are sorted in Table 4.2.

Area's colour	Area's function
Purple	Placing the prepared compounds
Green	Placing the finished solution
Orange	Operation
Red	Placing the heater
Blue	Placing the pool

Table 4.2: The function introduction of different areas in concept 2

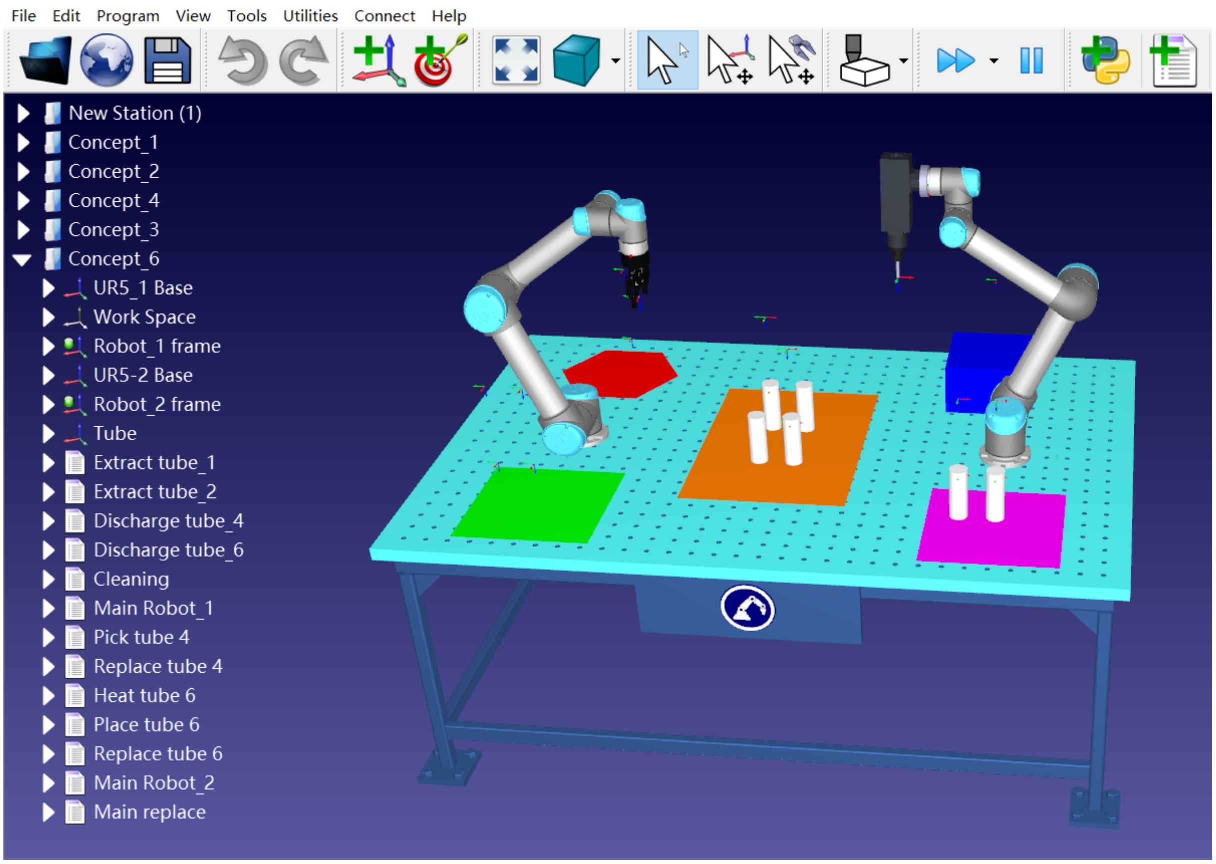


Figure 4.6: The simulation for concept 2

The advantages of this concept are:

1. There is no need to change tools during an operation to avoid a lot of trouble;
2. Make full use of the whole desktop than concept 1 to reduce regional waste;

The disadvantages are as follows:

1. High cost. The cost of two robots will double compared with a single robot;
2. The programming is more complex, so it is necessary to achieve the cooperative operation of two mechanical arms and control the time more strictly.

The advantages and disadvantages are sorted in the following Table 4.3:

Advantages	No tool replacement	Large working range
Disadvantages	High cost	More complex program

Table 4.3: The advantages and disadvantages of two robot concept

In the conceptual simulation, the simulation program is compiled in this project, which can completely simulate the dilution experiment of quinine solution. The final analogue video effect can be downloaded and viewed in GitHub:Automatic-synthesis-of-Molecules-by-UR5.

4.5 Conclusion

As the UR5 robot is used in this project, the scope of work will be narrowed compared with the chemical laboratory of AAU. In the future, researchers can consider replacing a giant robot, such as UR10 shown in Figure 4.7. At the same time, the placement position of desktop containers can also be enlarged in equal proportion. The disadvantages of a single robot are easy to overcome. However, the cost of the two robot concept is too expensive and does not have any advantage in terms of cost performance. Therefore, after comprehensive consideration, this project decided to adopt the idea of concept I.

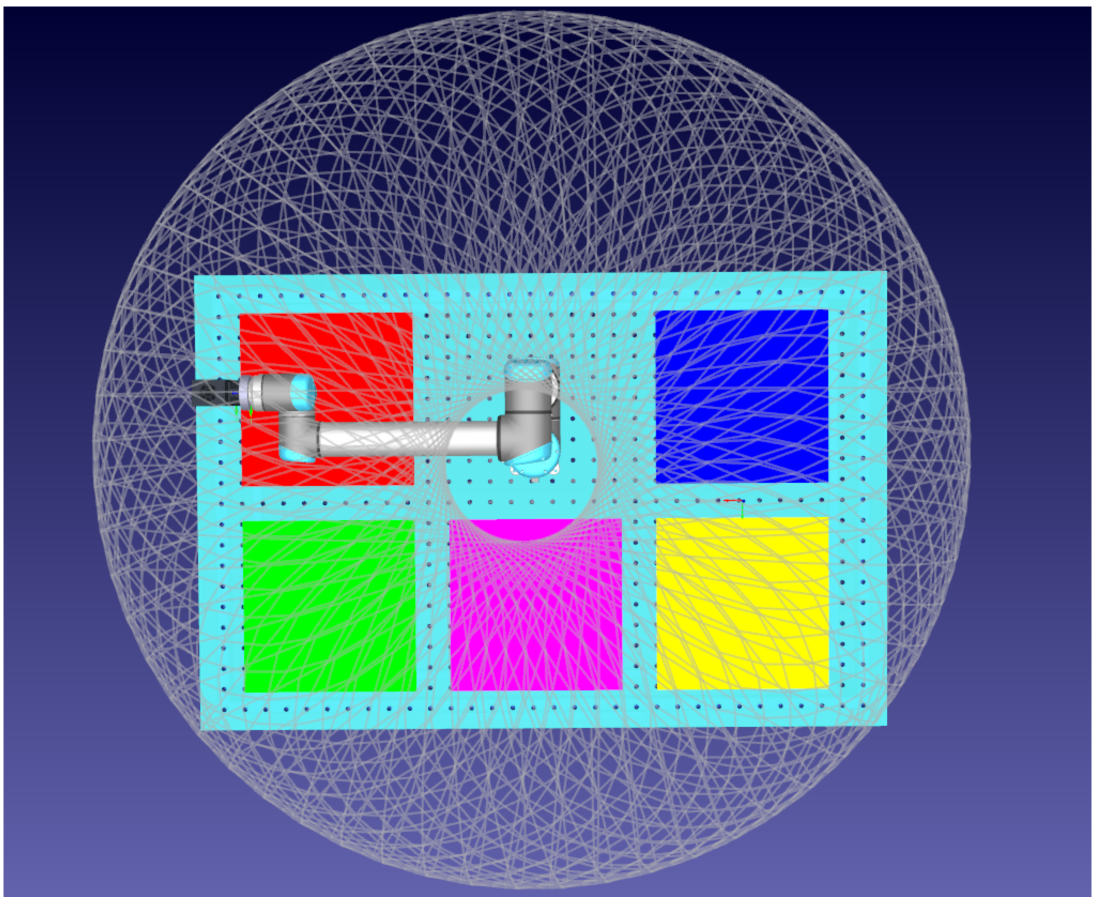


Figure 4.7: The work space that the UR10 can reach

Part IV

Practise

Chapter 5

Practise

This chapter will focus on restoring the simulation results to the physical environment and the problems that will be encountered.

5.1 Establish simulation environment

In order to program the robot more accurately and efficiently and to enable non-robot professionals to master and write the main program quickly, the project determines to simulate in the RoboDK environment that can automatically generate URscript and send it to UR5. The first part is to establish a simulation environment in RoboDK, and only on this basis can the desktop setting and the path planning of the robot be completed.

First, there must be an accurate UR5 robot model in the simulation environment. RoboDK has prepared many mainstream robot arms for R & D staff in its online library. Among them it has the robot arm used in this project. The addition method is: Please click the logo shown in Figure 5.1 to enter its online library directly. Directly search the '**Universal Robot**' in brand classification. Choose '**UR5**' and open it in RoboDK.

Secondly, establish a worktable (2m×1.2m) that meets the standards of the chemical laboratory. Since the worktable will not impact the final effect, two standard worktables (with a size of 1m×1.2m) of the online library are directly selected and spliced in parallel. The reason for choosing two small tables instead of a whole table is that the physical environment in the laboratory is only half of the standard. This advantage is that it can make the workspace more intuitive by hiding the second table in subsequent experiments. At the same time, if needed in future experiments, it can be quickly modified by 'displaying' the second table. (the displayed operation is to right-click the object to be displayed/hidden and select 'visible')

Finally, to simulate the effect of the closed environment of the workbench, the cell frame in the 'tool' can also be added. The final effect is shown in Figure 5.2. However, to more clearly observe the movement of the manipulator and the changes on the worktable, it is hidden 'in subsequent operations. So far, the initial establishment of the simulation environment has been completed.

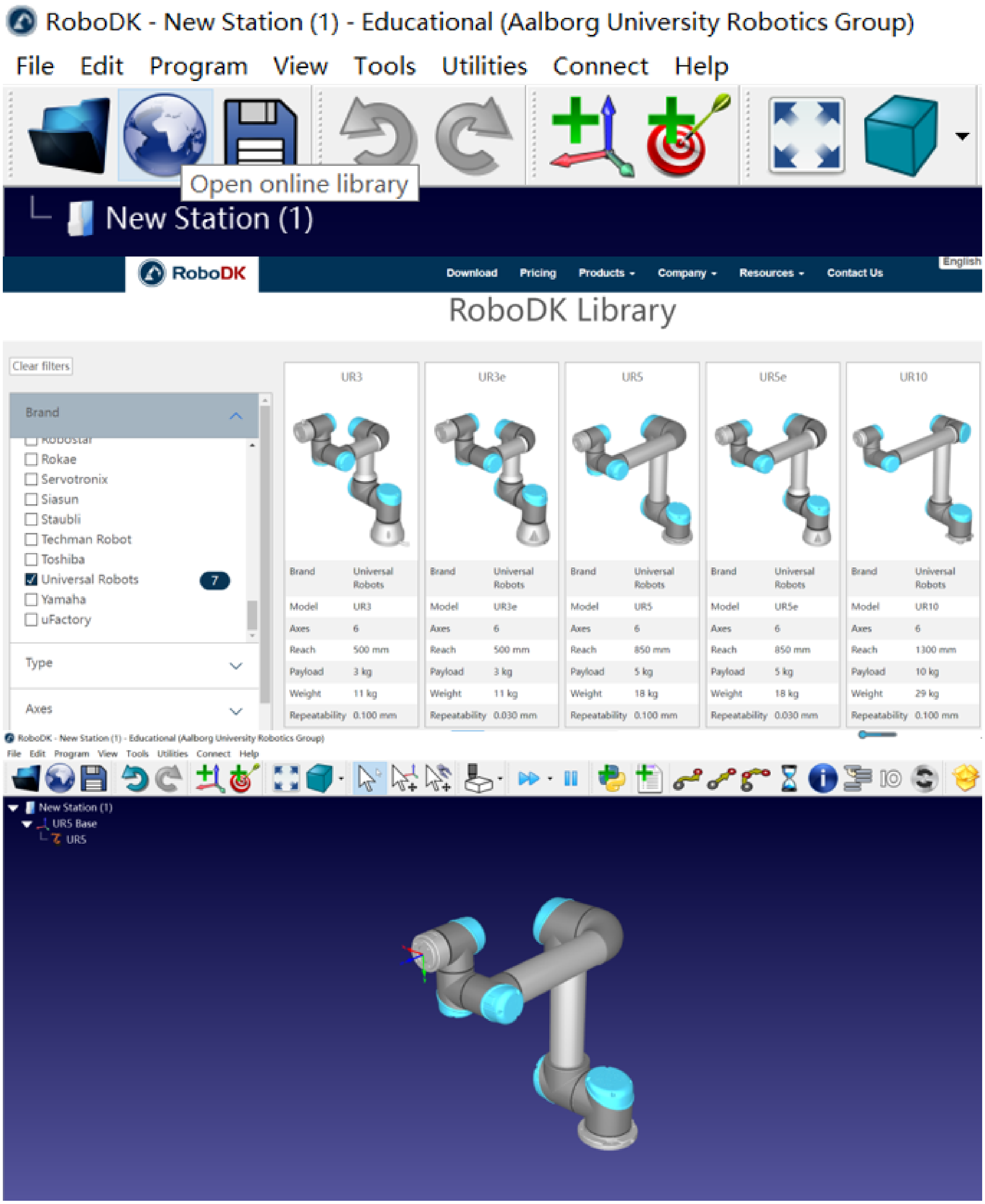


Figure 5.1: How to choose a robot in RoboDK

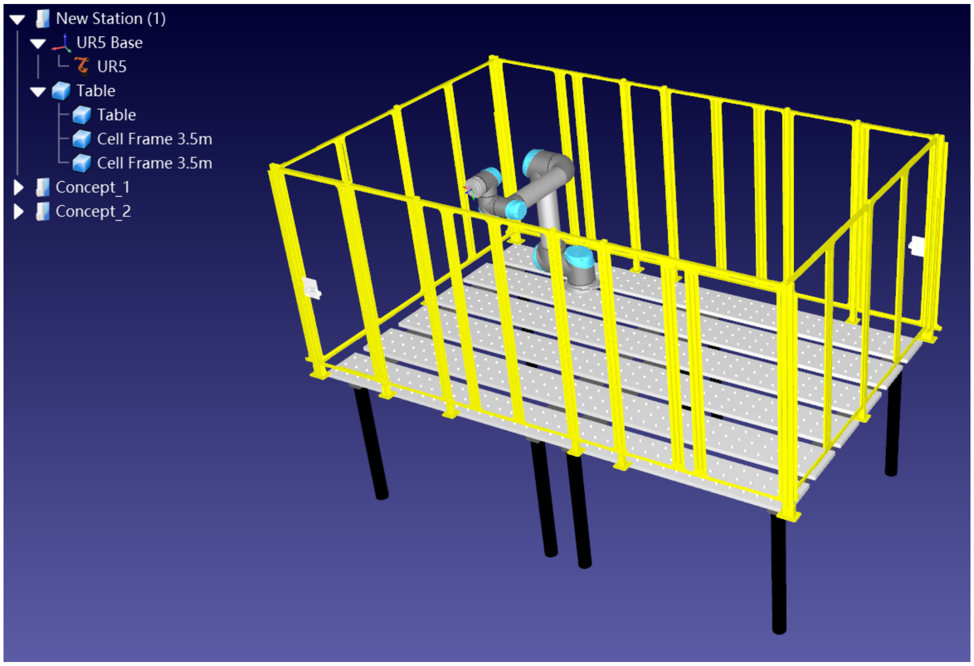


Figure 5.2: The basic robot cell in RoboDK

5.2 Set coordinate system

Because the robot system needs to use many coordinate systems, such as tool frame, robot base frame, world frame, and so on, this section will introduce all coordinate systems used in the project and their functions.

5.2.1 Robot base frame

The robot base frame is also the reference frame, which refers to the coordinate system established by the whole system with the base of the robot as the origin. The objects in this frame should be directly related to the robot, so it should include: the robot, Syringe, and Gripper. The specific parameters are shown in Figure 5.3:

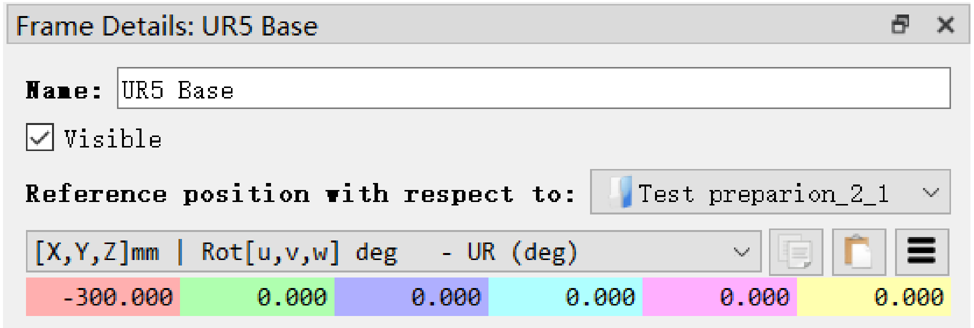


Figure 5.3: The parameters in the UR5 base frame in RoboDK

5.2.2 World frame

The world frame is used to place objects that do not affect the program, for example, workbench, cell frame, test tube fixture, heater, and test tube. To quickly and directly set the path points of the robot action, it is not necessary to calculate the actual coordinates through the coordinate transformation matrix. The parameters of the world frame should be consistent with those of the reference frame.

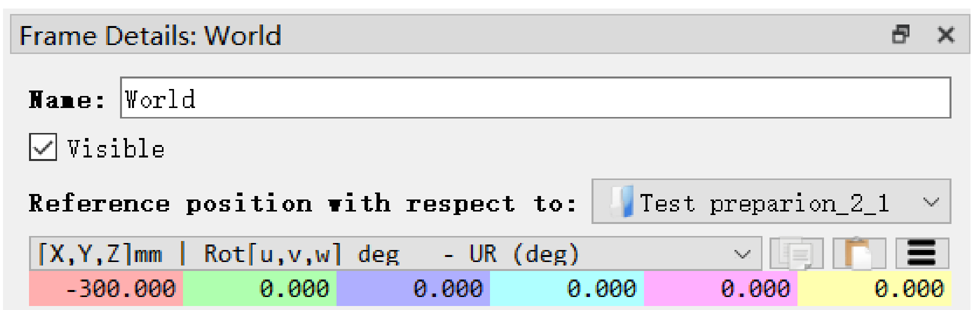


Figure 5.4: The parameters in the world frame in RoboDK

5.2.3 Working frame

The working frame is the frame used to place the robot way-point. In robot programming, the destination is usually set as the way-point to clarify the specific position reached at the end. The line automatically planned by the program between the way-points is called the moving path of the robot. Therefore, an independent frame is needed to place each required way-point. Also, the selected parameters are consistent with the datum frame for the convenience of coordinate processing.

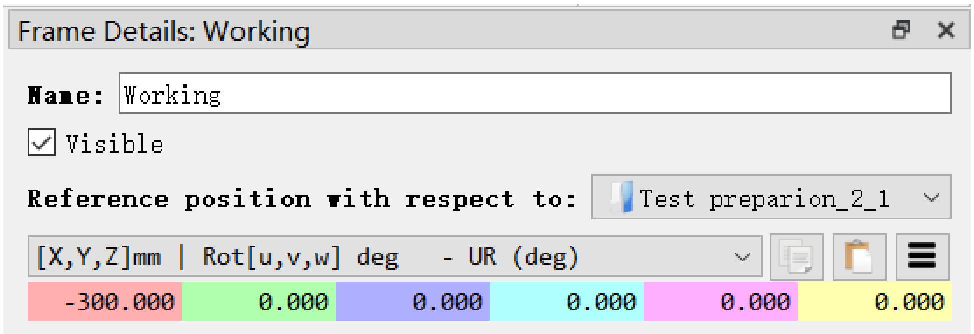


Figure 5.5: The parameters in the working frame in RoboDK

5.2.4 Tool frame

The tool frame is established with the centre point at the end of the tool as the origin. Since there is no intuitive connection between the datum frame and the set road points, the frame of the point in direct contact with the way-points will be established. The advantage is that adjusting the robot's position can intuitively judge whether the position is the target position. However, it should be noted that the tool frame is one of the frames of the robot and does not store any object or point position. The way-points data will be placed in the working frame instead of the tool frame after data processing.

After each tool is installed, its tool frame needs to be reset. Therefore, how to modify the tool frame will be described below. There are two ways to set up 'tool centre point'(TCP) on RoboDK, and both methods have taken syringes as the example.

The first method is to drag TCP on the image directly. Click the icon shown in Figure 5.6 in the RoboDK interface. At this time, the tool frame is highlighted, as shown in Figure 5.7. Just drag TCP directly to the target position. This method is simple and straightforward to use, but its disadvantage is also pronounced. The position is not accurate enough and can usually be used as a coarse adjustment. For precise adjustments, please refer to the other methods.

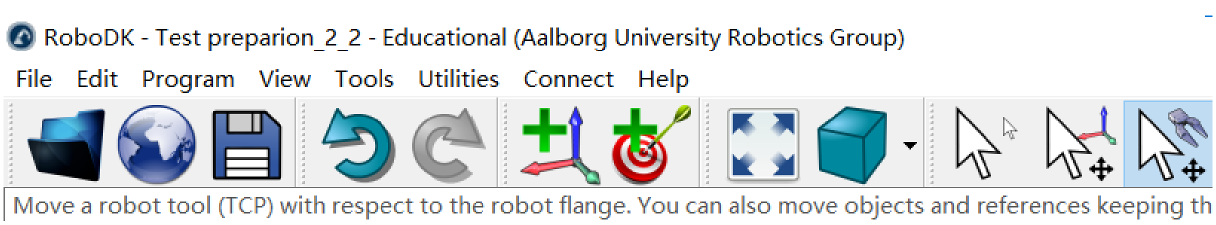


Figure 5.6: The parameters in the working frame in RoboDK

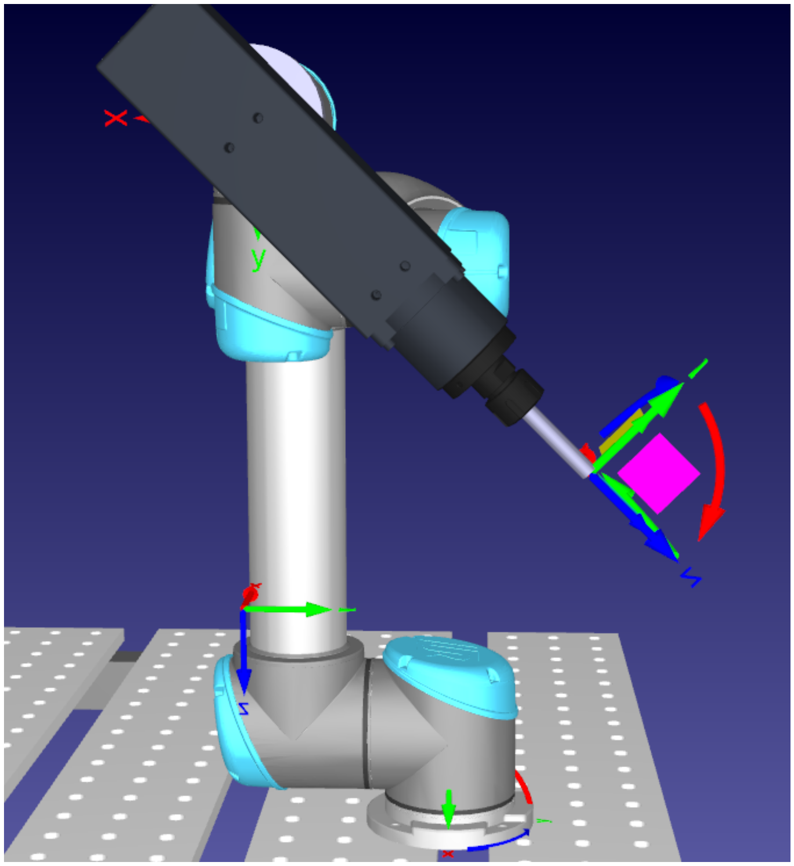


Figure 5.7: The parameters in the working frame in RoboDK

The second method is calculated through data, which is more accurate and suitable for the current project.

- Step 1: set a certain point in the plane. For example, fix a pen on the workbench. The nib is used as the determination point to lay a good foundation for the follow-up;
- Step 2: as shown in Figure 5.8, click ’**Utilities**’ and’**Define Tool Frame (TCP)** ’to get the page shown in Figure 5.9;
- Step 3: manually control the robot to align the end of the syringe with the nib. Find the position data; under this posture on the robot teaching pendant;
- Step 4: input the data into the first row of the window shown in Figure 5.9;
- Step 5: manually control the robot and change the angle to align the end of the syringe with the nib again. And input the data on the teaching pendant into the second line again;
- Step 6: repeat step 5 twice to obtain the complete posture data of the four robots;

- Step 7: click '**Update**' to complete TCP adjustment;

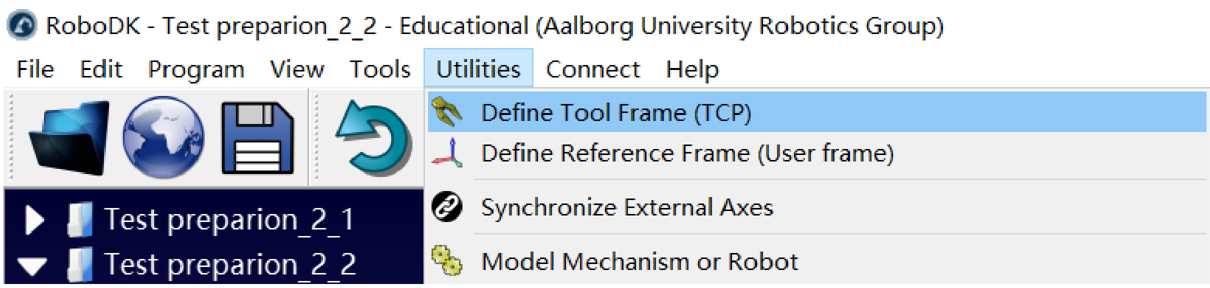


Figure 5.8: How to set the TCP in RoboDK

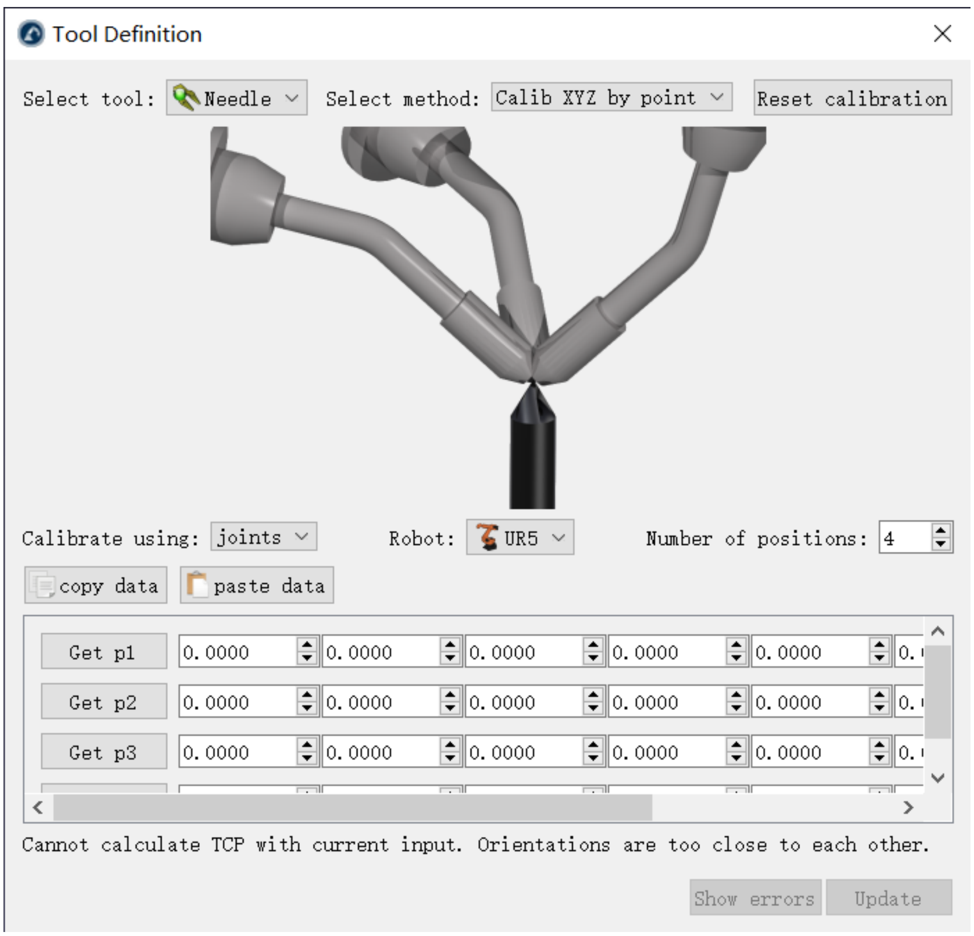


Figure 5.9: The tool definition window in RoboDK

The completed tool frame parameters are shown in Figure 5.10. After adjusting the TCP of both gripper and syringe, the setting of all frames is completed.

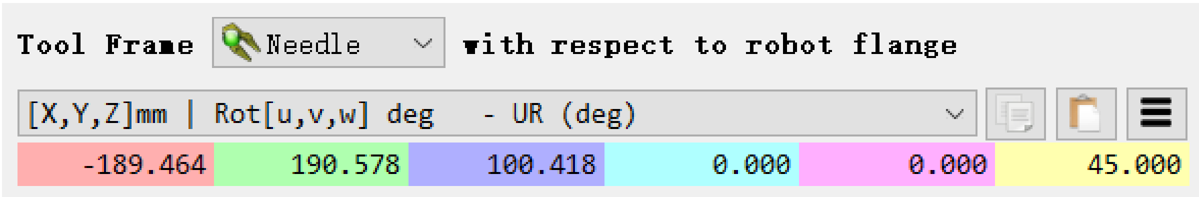


Figure 5.10: The parameters in the tool frame in RoboDK

5.3 Dimension measurement

Firstly, it needs to be clear that not all items used in this project are standard parts, which models can be downloaded directly through the online material library. Non-standard parts here specifically refer to objects with the following two types of characteristics: 1. When the robot sets the way-point, the size of the object needs to be used as a reference; 2. The specific shape and size of the object cannot be downloaded from the open-source material library on the network. The dimensions that need to be referenced for the robot path planning in the first point are critical dimensions. According to the understanding of the experimental process, the following key dimensions can be found:

- The diameter of the syringe is used to judge whether the syringe can be extended into the test tube;
- Diameter of the large test tube;
- Height of the large test tube;
- Diameter dimension of the large test tube fixture;
- Height and depth of large test tube fixture;
- Diameter of the large test tube;
- Height of the large test tube;
- Diameter dimension of the small test tube fixture;
- Height and depth of the small tube fixture;
- Outer ring diameter dimension of heater;
- Diameter dimension of heater inner holes;
- The number of holes in the heater and their position distribution;
- Maximum travel of gripper;

- Distance from the end of the gripper to the robot; flange in three standard coordinate axes (x-axis, Y-axis and z-axis directions);
- Distance from the end of the syringe to the robot flange in three standard coordinate axes (x-axis, Y-axis and z-axis directions);

Based on the above critical dimensions, it can be determined that the object with the first feature includes a syringe, gripper, large test tube, small test tube, large test tube fixture, small tube fixture and heater. The gripper belongs to the robot tool circulating in the market and can be downloaded to the three-dimensional model in GitHub:Automatic-synthesis-of-Molecules-by-UR5. Therefore, the gripper is not a non-standard part of the project, and its related dimensions do not need to be measured. After measurement, non-standard parts and their key dimensions can be sorted into the following standard two-dimensional views, which are shown in following figures.

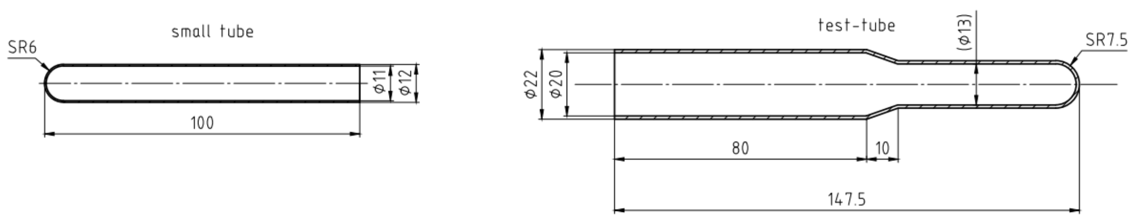


Figure 5.11: The 2D three view of the test tubes

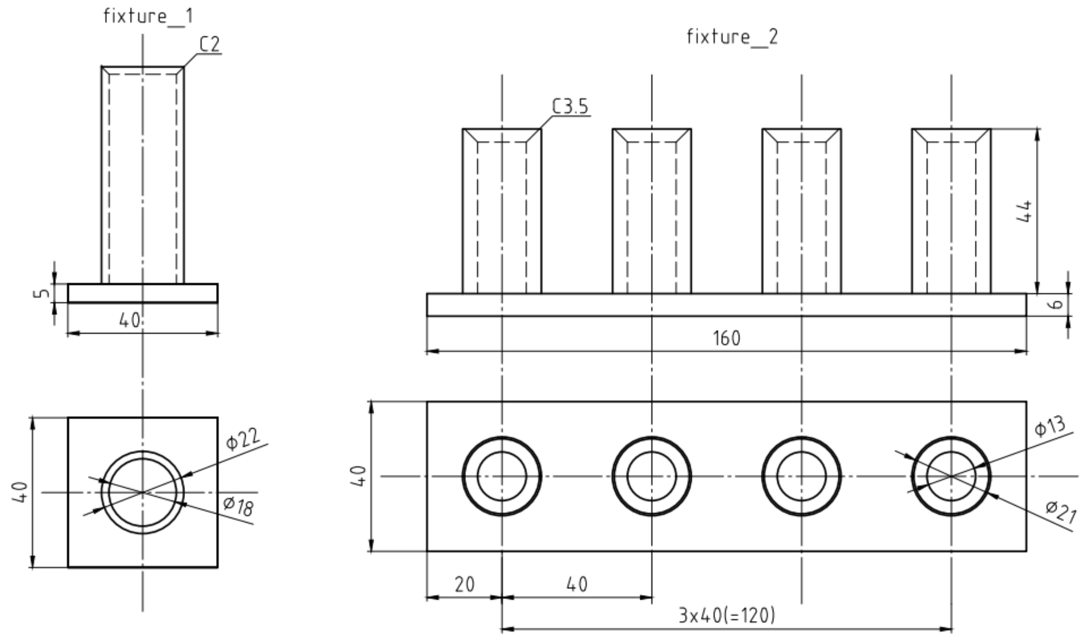


Figure 5.12: The 2D three view of the fixtures

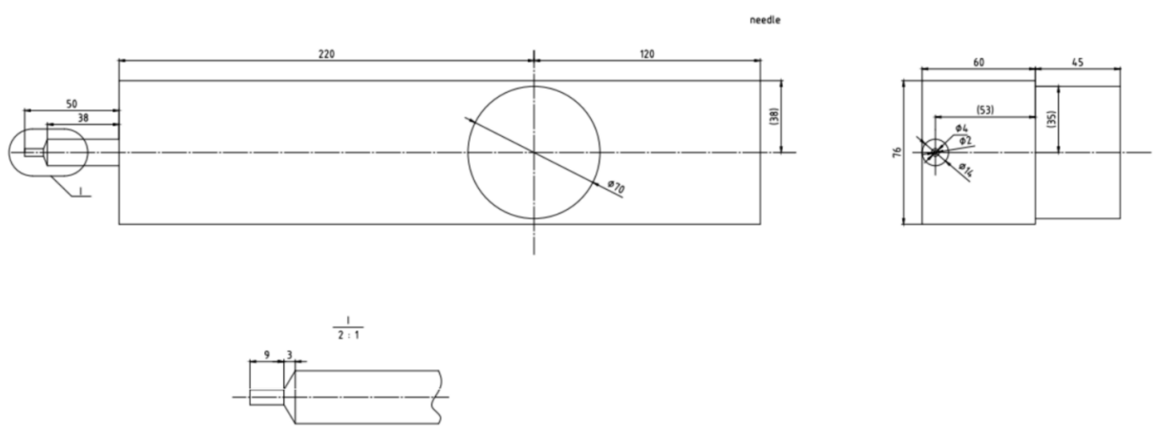


Figure 5.13: The 2D three view of the syringe

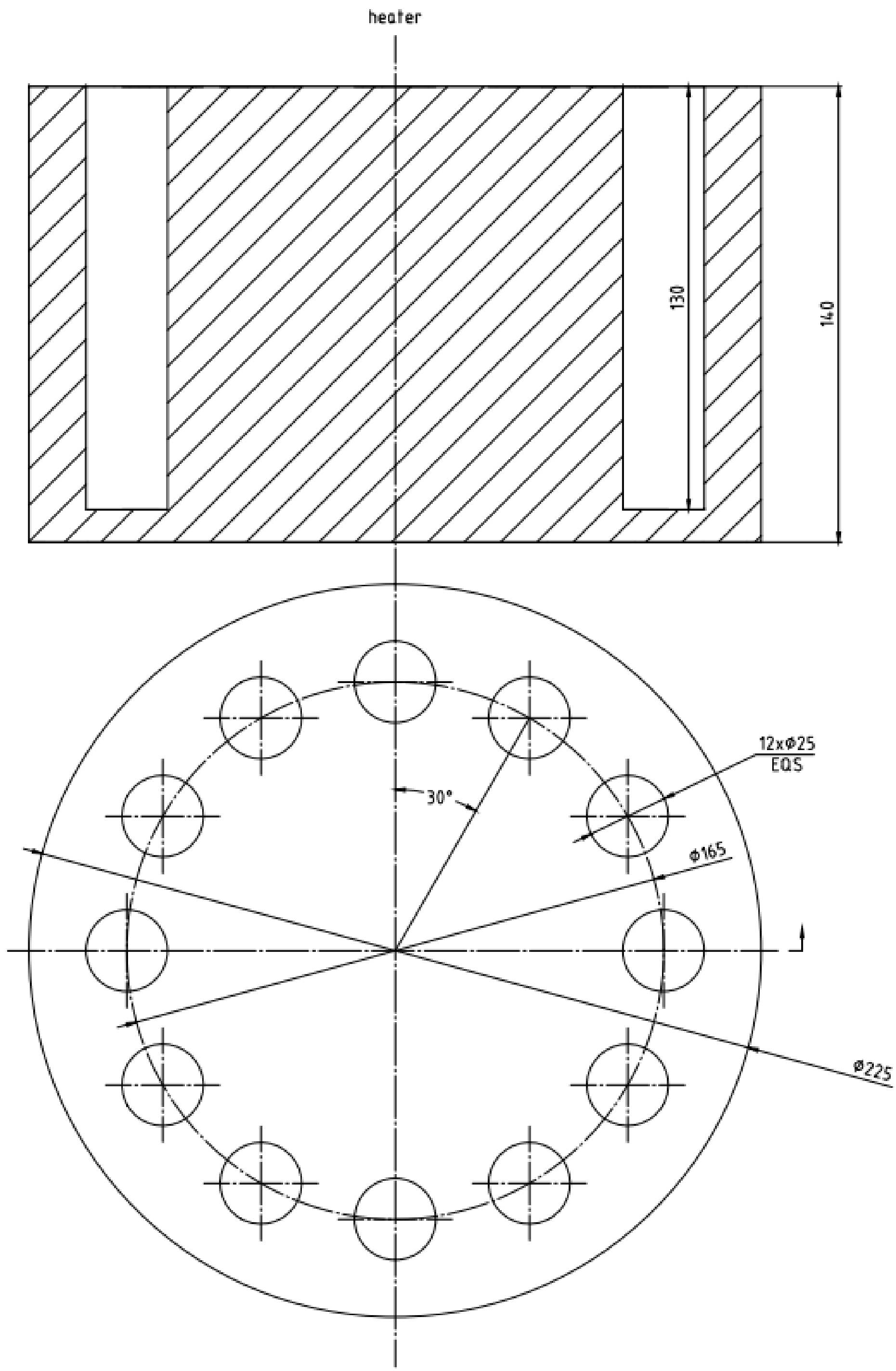


Figure 5.14: The 2D three view of the heater

5.4 3D modelling

3D modelling is a necessary step to fully reproduce the most accurate critical size data of tools and containers in the RoboDK simulation environment. Refer to section 5.3 for all non-standard objects that will affect the setting of way-points. They are syringe, test tube, small tube, test tube fixture, small tube fixture and heater. The 3D modelling of these objects will be carried out according to the dimensional data measured in the previous section. There are two points to note:

1. This model is only the basic model to facilitate the robot to set the way-points. It only contains all key dimensions without perfect restoration of their shape;
2. Creo 4 software is adopted in this project to draw 3D models, and the output format must adopt **STL or STP**;

Finally, the following 3D model will be drawn according to the size measured in the previous section and can be added to the simulation. But it is worth noting that unit conversion is required if operators use these four files named ’Fixture_ 2’, ‘ Heater ’, ‘ Syringe ’and ’Small tube’. The method is to double-click the object and modify the size to 0.03937 times in the right attribute bar.(1mm=0.03937inch)

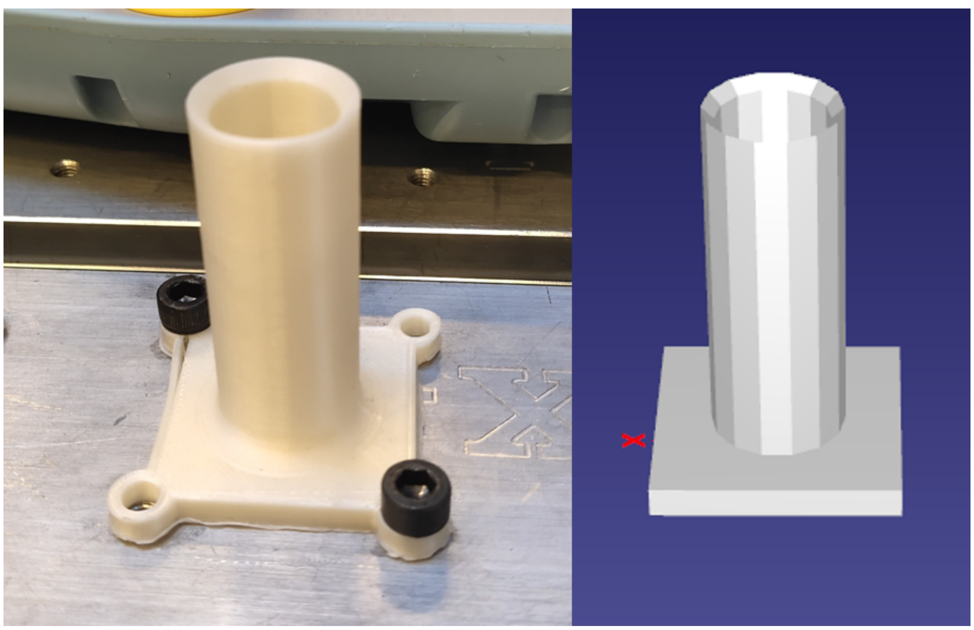


Figure 5.15: The 3D model of the fixture_1

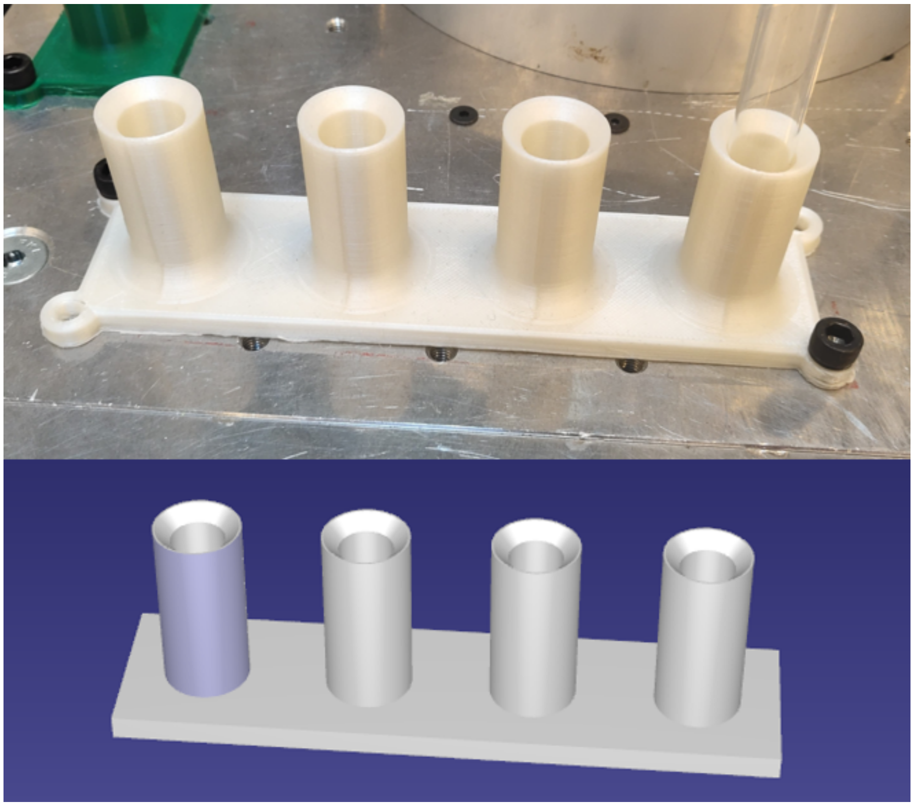


Figure 5.16: The 3D model of the fixture_2

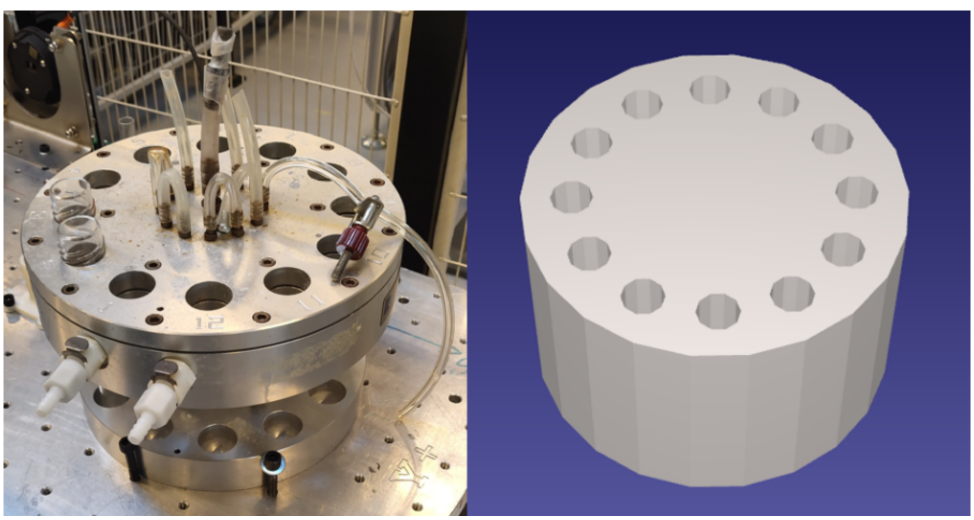


Figure 5.17: The 3D model of the heater

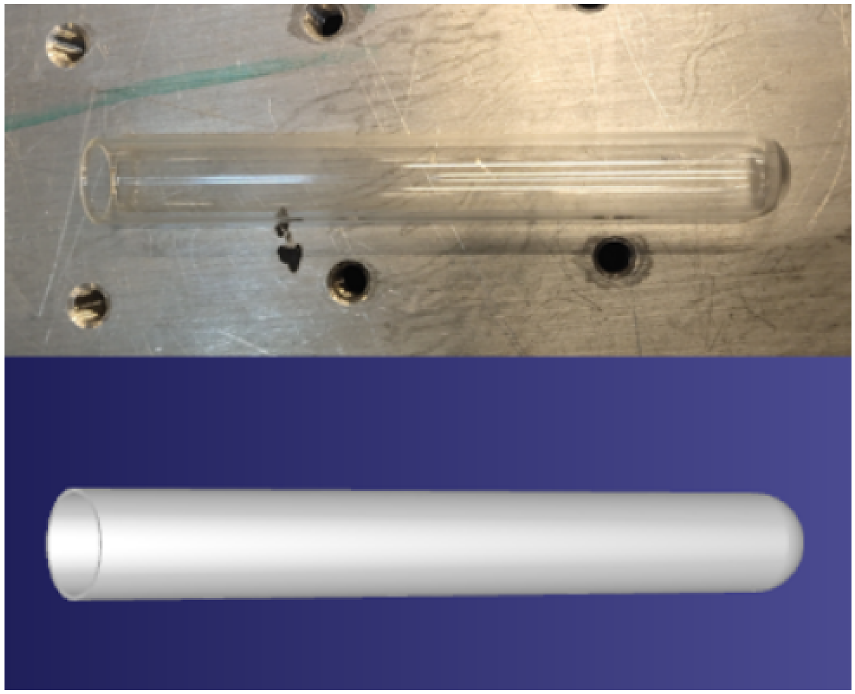


Figure 5.18: The 3D model of the small tube

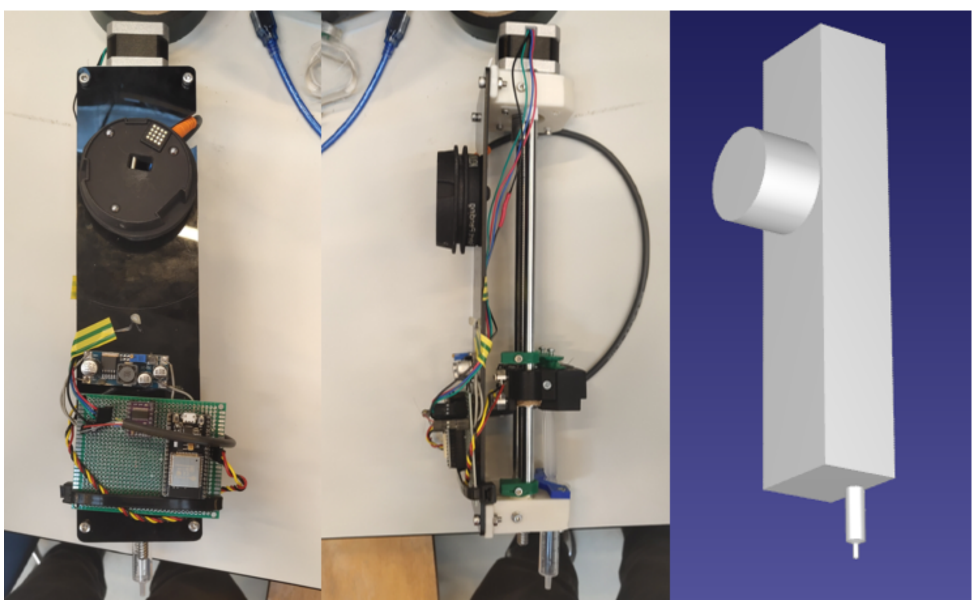


Figure 5.19: The 3D model of the syringe

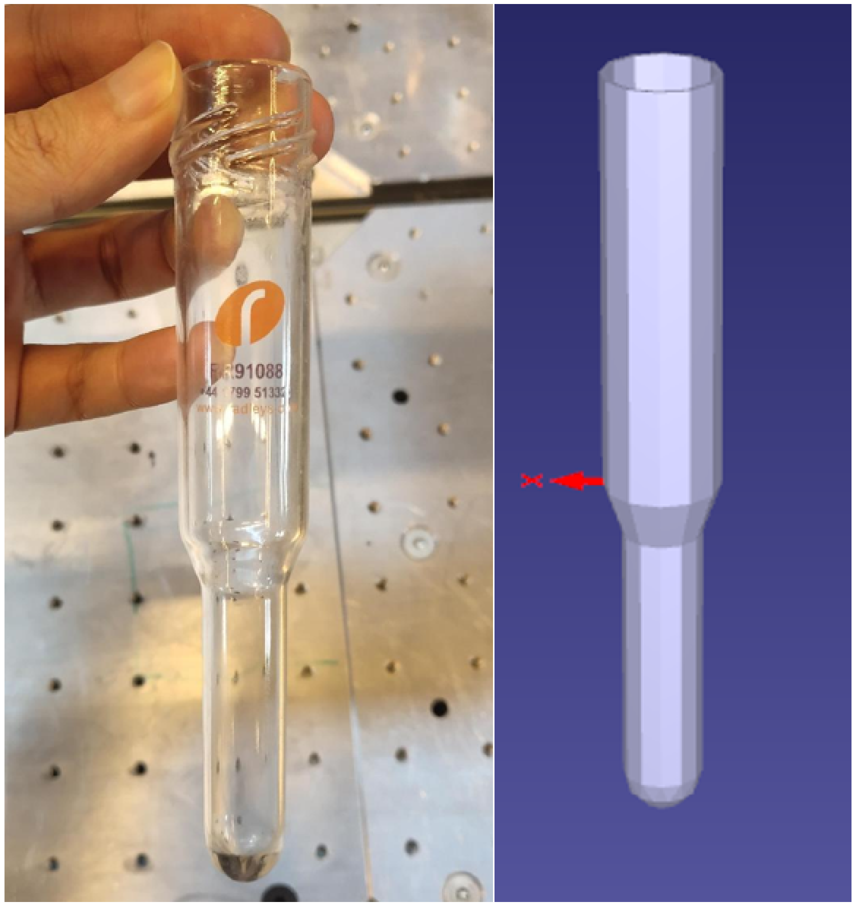


Figure 5.20: The 3D model of the test tube

5.5 Restore the location of the real environment

To correspond the objects' position in the simulation with reality, this project takes the UR5 robot as the zero point to measure the coordinates of other objects. The layout of all objects meets the project requirements. All coordinate data are accurately recorded in the software workstation. For specific values, refer to Figure 5.21 or download the workstation in GitHub:Automatic-synthesis-of-Molecules-by-UR5. Because the number of test tubes and fixtures is insufficient, the division of preparation area, operation area and completion area is not reflected. However, this method can still be followed in future experiments to rebuild the workbench and write programs quickly.

Operation method: double click the selected object, click '**more operations**' in the right attribute bar, and enter the coordinate value at the lowest position parameter.

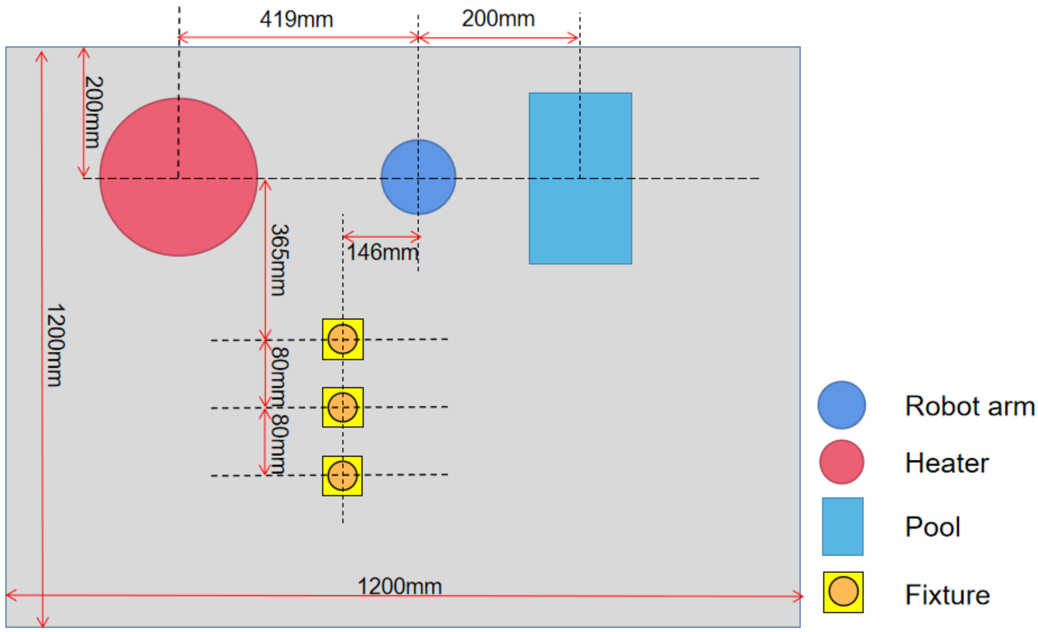


Figure 5.21: The dimension of the workbench's layout

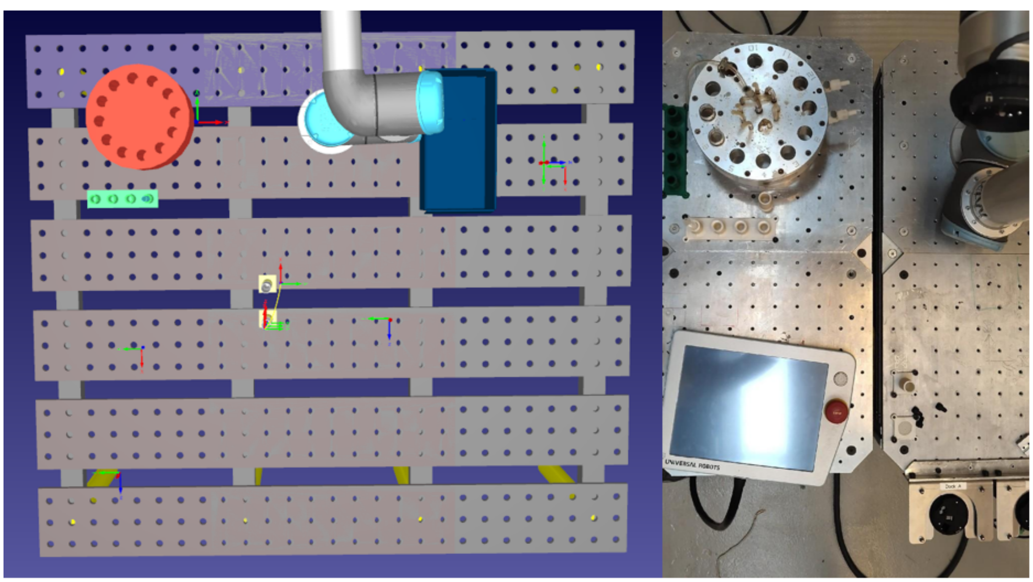


Figure 5.22: The simulation and physical layout of the workbench

5.6 Path planning

Path planning is to set several way-points based on the working coordinate system at the robot's end. The program can freely assign whether a straight line or curve will connect the way-points. The line between points is called the moving path of the robot. The robot's path planning is how to arrange the sequence of these way-points and the connection type between points.

The first thing to be confirmed is which points are worth setting as way-points. Taking the gripper as an example, the positions where all test tubes need to be clamped need corresponding way-points. When the gripper moves to these way-points, it means that the opening and closing of the gripper can accurately control the corresponding test tube. These way-points can be called working points. Secondly, vertical space is needed as a buffer before the robot moves to the target position. Therefore, a buffer point, also known as the preparation point, is required directly above all clamping points at a certain distance. Usually, a constant Z value should be set in the robot's frame to correspond to the desired height. In addition, the robot's home position needs a home point, and a starting point of action is also required directly above the working area. Since the length and end position of the syringe are different from those of the holder, the syringe needs to be reset to another set of points. The position of the starting point and the home point remains unchanged. The working point becomes the coordinate where the syringe goes deep into the inside of the test tube. The preparation point needs to be adjusted with the length of the syringe. Cleaning points shall also be set above and in the pool's centre. Setting way-points is: move the robot to the required posture, select the working coordinate system and click the icon shown in Figure 5.23 to create the corresponding target. At this point, the way-point is created. So far, all way-points have been set.

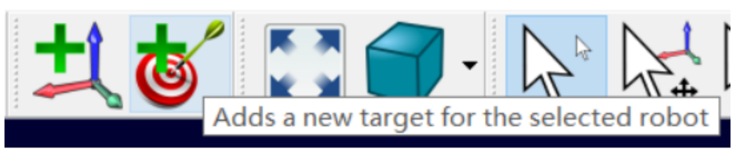


Figure 5.23: How to add a target as the way-point in RoboDK

Then make a program of basic actions with high repetition. Create a program by right-clicking 'Simulation Event Instruction' in '**Add Instruction**'. Select '**Set Object Position (relative)**' in '**Action**' and select both robot frame and working frame. Select the corresponding way-points in the order of the desired moving position of the robot, and finally choose the type of the path as shown in Figure 5.24, i.e. linear or circular. The program has high flexibility, so it needs to be arranged according to the experimental needs.

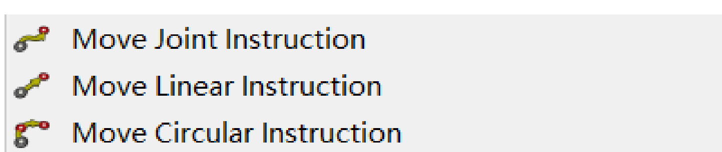


Figure 5.24: Three path types can be selected as required

5.7 Control of tools

The two common tools used in this project are the gripper and the syringe. Because the syringe controlled by the robot is not a commodity that can be purchased in the market, AAU designed a physical structure and a control system for the standard medical syringes. The motor's output is controlled through the Arduino board, and the motor rotates to drive the screw rod to move up and down, which finally completes the control of the syringe. The structure can be seen in subsection 2.3.3. This section will explain how to control these two tools and write them into the main program.

5.7.1 Gripper

There are two methods to control the gripper. First of all, the standard method is to use the written program for control. According to the system requirements of the UR5 robot, Python needs to be used to write URscript code and pass the instructions to the robot through a communication protocol. Note that this program is an open-loop control[14]. Hence, the program only sends the command to control the gripper but does not receive feedback information from the robot and will not make modifications at the program level according to the feedback results. However, this does not mean that the gripper is uncontrollable. Because both UR5 and RG2 have force feedback sensors, the robot will make flexible operations according to the feedback force after clamping. When the pressure obtained by the force sensor reaches the preset critical value, the gripper will not continue to tighten. The program flow chart is shown in Figure 5.25.

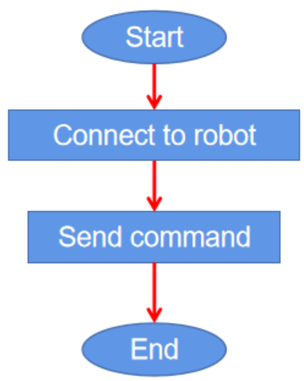


Figure 5.25: The flow chart of gripper control

According to the flow chart in Figure 5.26, the following code can be compiled. The specific code can be downloaded at GitHub:Automatic-synthesis-of-Molecules-by-UR5:

The first method's disadvantage is that it can only be used alone. The program cannot be compiled into the main program and the error of "**unable to find the program**" will be reported. Therefore, it cannot be used in the main program of RoboDK. But the advantage is that Python can directly control it. The platform is not limited, and the program can be run as long as Python can be run.

```python
from robolink import *    # RoboDK API
from RoboDK import *      # Robot toolbox
import socket
import time

RDK = Robolink()
# IP-addr/ports for UR
UR_IP = '192.168.0.102'
UR_PORT = 30001
def connect_UR():
    ur_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    ur_socket.settimeout(2)
    ur_socket.connect((UR_IP, UR_PORT))
    return ur_socket
def send_cmd(ur_socket, cmd):
    cmd = cmd + '\n'
    received=ur_socket.send(cmd.encode())
    received = ur_socket.recvfrom(4096)
    return received

def main():
    ur_socket = connect_UR()
    result = send_cmd(ur_socket, "set_tool_digital_out(0, False)" + "\n")
    time.sleep(1)
if __name__ == "__main__":
    main()
```

Figure 5.26: The Python code that controlled the gripper [15]

Another method is to call the I/O digital output signal in RoboDK. The specific operations are: create a new program, right-click '**Set or Wait I/O Instruction**' in '**Add Instruction**', as shown in Figure 5.27. Set the port to '**10**'(which may be different from other robots), and the digital output

'0' represents opening the gripper. In contrast, the digital output '1' represents the closing gripper. The advantage of this method is that it can be freely added to the main program without compilation problems. The disadvantage is that the platform is limited and can only be used in RoboDK.

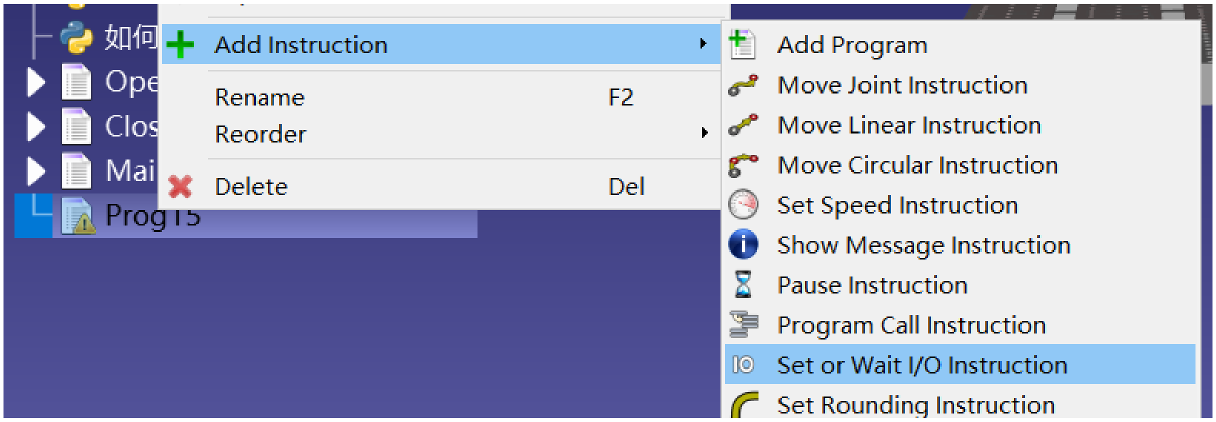


Figure 5.27: How to set the digital output in RoboDK

Finally, the program of the gripper is completed. The effect of the workstation should be the same with that in Figure 5.28

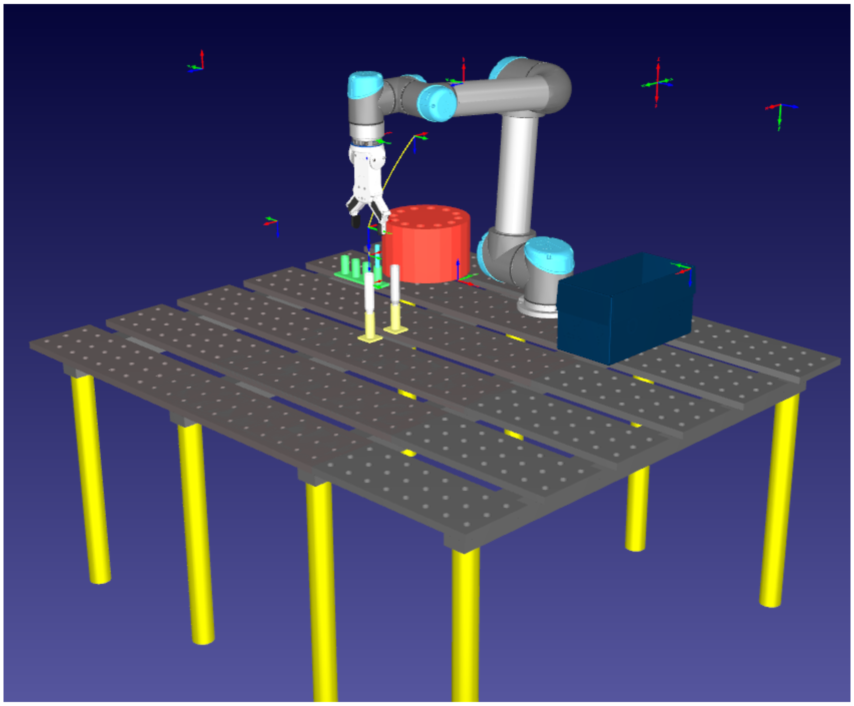


Figure 5.28: The final effect of the gripper's workstation

5.7.2 Syringe

Syringe control is the part that cannot be realized in this project. The reason is that the control of the syringe is independently controlled by Arduino and independent of the mechanical arm. The software can only simulate the output of the robot. Therefore, the software can not drive the motor of the syringe directly. At present, the method used by the AAU laboratory is to transmit commands through the MQTT communication protocol. The essence is to write C + + code that can control Arduino IDE on the syringe and burn the code into the Arduino board through the data line port. The execution command is sent to the subscriber of the syringe through mosquitto so that it can execute the forward or reverse rotation of the motor. The specific principle can be shown in Figure 5.29, and the C + + code can be viewed and downloaded at GitHub:Automated_synthesis[16].

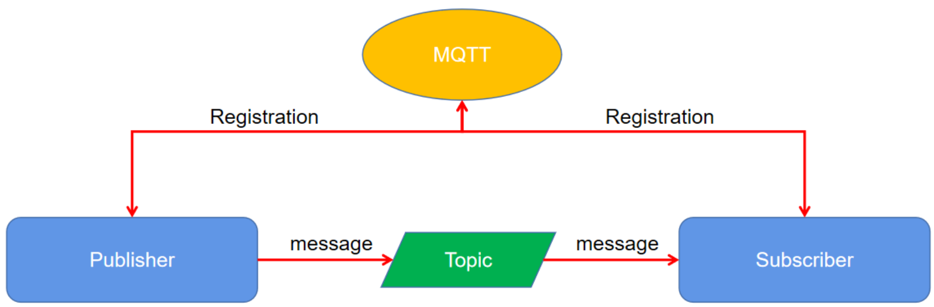


Figure 5.29: The working principle of the MQTT communication protocol

The above method cannot achieve the goal since RoboDK supports Python programs and does not support C++ programs. Here are two possible approaches to controlling the syringe, which can be further studied in future work:

First, the first method does not need to change the syringe structure. But under this premise, hundreds of C++ code lines must be rewritten into Python code. Since Arduino usually does not support Python, the solution is to compile and upload the Firmata package to the used Arduino in advance so that the control board can execute Python code. The successfully verified code can be added in a method similar to controlling the gripper in subsection 5.7.1. The advantage of this method is that there is no need to modify the structure of the syringe, but the disadvantage is that Python code cannot be typically written into the main program. The principle is that when the software sends the main program to the robot, it will automatically generate URscript code. However, the code will only display the name of the python program, but the content will be missing, so the system will report an error. When the syringe needs to execute the command, the operator must manually double-click the corresponding Python program.

The second method needs to change the syringe structure and bind the trigger condition of Arduino to the output of the robot. Since the syringe does not need to consider the stroke length of push and pull (for this reason, please refer to the description of the syringe principle in subsection 2.2.3), the digital output can be selected as the trigger signal. Set the IO value at a specific port through the event simulation of RoboDK, and the program can be directly written into any main program. When the robot receives the main program, the terminal outputs current, which is also used as the input signal of Arduino to trigger it to execute the corresponding program. So far, the control operation of a set of syringes is completed. This method has the advantages of concrete universality, can adapt to the development of various tools and can be directly written into the main program without additional control of operators. The effect of the workstation should be on the basis of that in Figure 5.30

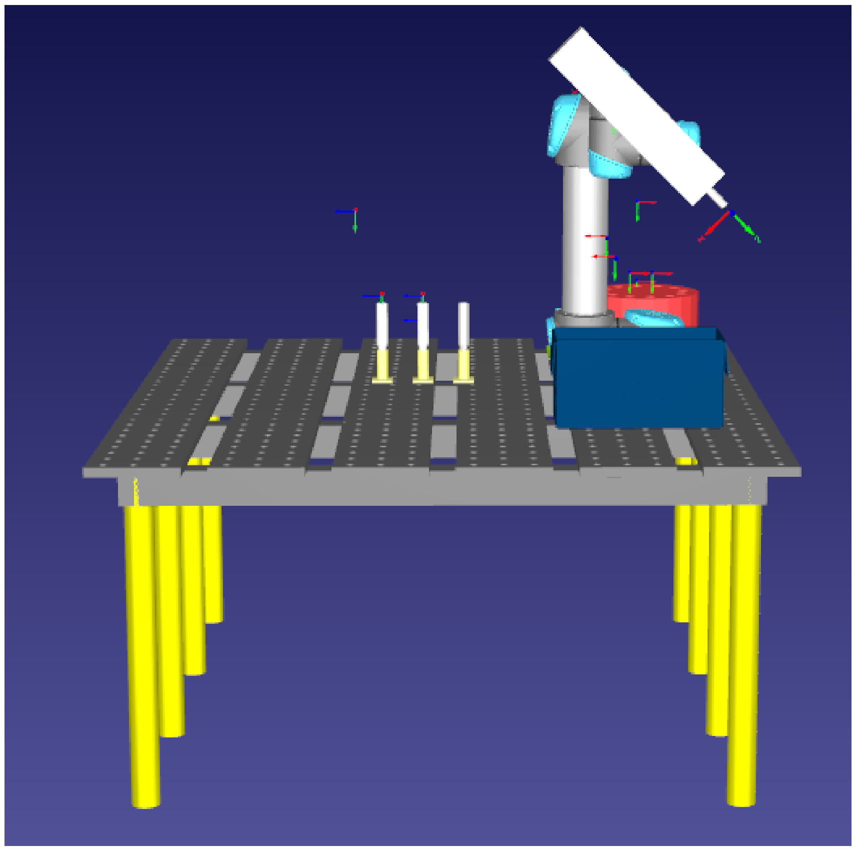


Figure 5.30: The effect of the syringe's workstation

5.8 Write the main program

The logic of the robot's action needs to be sorted out. This section will take the pickup and place of test tubes as an example to introduce the operation process in detail.

First, determine the experimental goal. That is, the robot needs to transfer the test tube from Fixture_2 to Heater_2 and can also reverse the operation. According to the description of the general task, the experiment includes the following basic actions: the robot returns to the initial point, the robot grabs the test tube from the Fixture_2, the robot places the test tube at the Heater_2, the robot grabs the test tube from the Heater_2, and the robot places the test tube at the Fixture_2. To ensure the modular continuity of the above five basic actions, each primary action's starting position and ending position are in the same position, which can also be regarded as the interface of the subprogram module. In this way, no matter whether any subprogram is spliced, the robot's actions can be seamlessly connected.

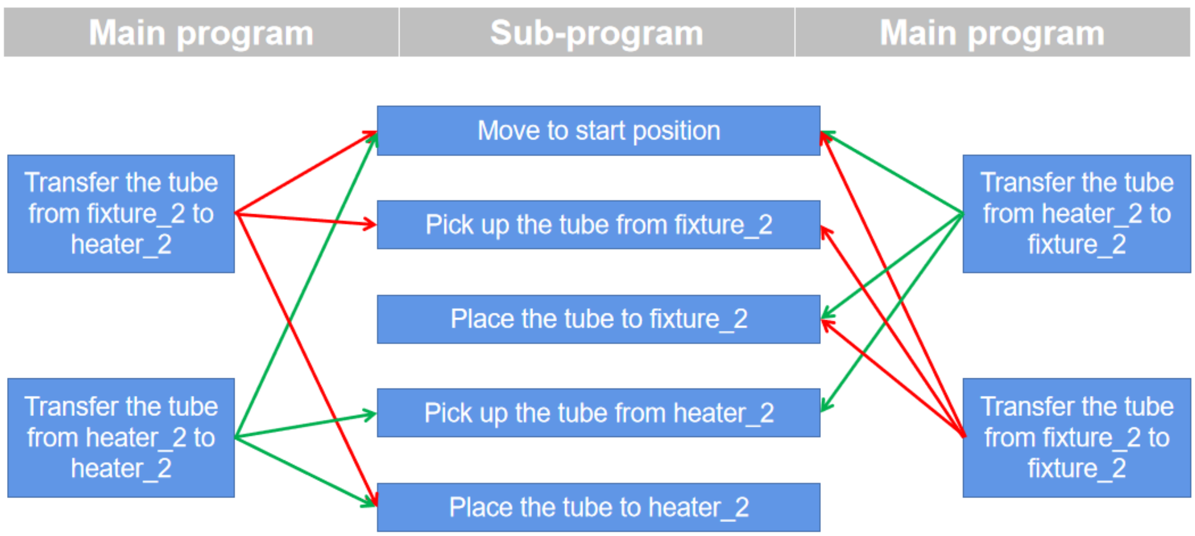


Figure 5.31: The logical relationship between main program and subprogram

In addition, to ensure that the robot's action is relaxed and will not cause damage to the equipment or test tube due to too fast speed, the program needs to increase speed control. The speed control principle is that when the gripper moves vertically around the test tube, the speed must be reduced to **50mm/s**. The full speed of the robot is **500mm/s**. The speed control method is: to right-click and select '**Set Speed Instruction**' of '**Add Instruction**' before the moving instruction is modified. Change the moving speed value in the'**Set Value**' window.

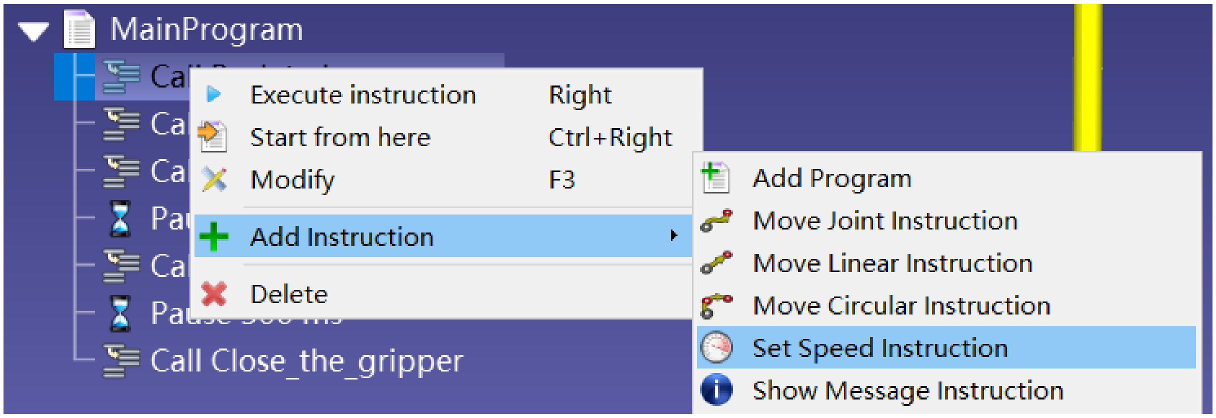


Figure 5.32: How to modify the speed of the robot

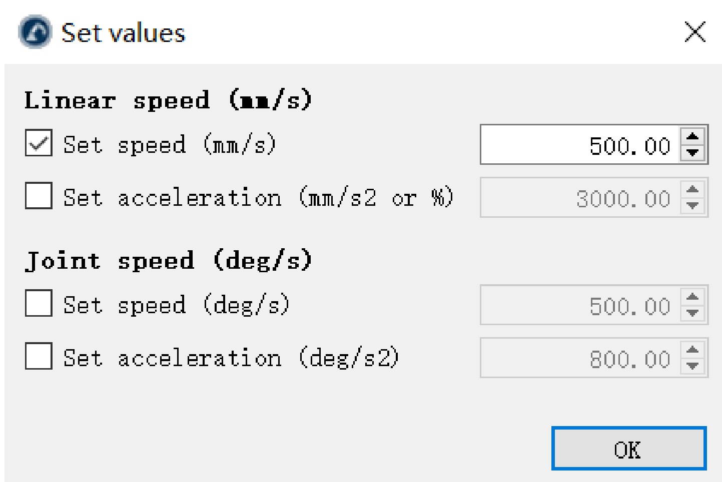


Figure 5.33: Modify the speed value in 'Set Value' window

Taking the "grabs the test tube from the Fixture_2" program as an example introduces how to write the subprogram. The first step is to move the robot to the starting point (full speed) and ensure that the gripper is open. The second step is to move the robot curve directly above the test tube for preparation. The robot descends vertically to the clamping position (slow speed) in the third step. Step 4: close the gripper. Step 5: the robot rises vertically to the ready position (slow speed). Return to the starting position at full speed (step 6). In this process, steps 1, 2, 3, 5 and 6 involve path planning. For the method, please refer to section 5.6. Step 4 involve the control of the gripper. Please refer to subsection 5.7.1 for the method.

According to the sequence of subprograms, right-click the prepared subprogram and select 'Make the Main Program'. To ensure the smooth operation of the main program, the waiting time instruction should be added as a kind of adjustment. So far, the main program has been written. The prepared workstation can be downloaded in GitHub:Automatic-synthesis-of-Molecules-by-UR5.

5.9 Test verification

After writing the main program, you can try to send the program to the robot and test and modify the program through the actual operation of the robot. Connecting the software to the robot will be performed as the first step. First, select ’**Network Setting**’ in the teaching board of UR5 and record the IP address of the robot. Secondly, in RoboDK, click the "**Connect Robot**" option as shown in Figure 5.34. Enter the value shown in Figure 5.35 in the left status bar. Note that the IP address of the robot may have been changed, so operators need to check whether it is consistent before entering. After everything is set, click ’**Connect**’ to complete the operation.

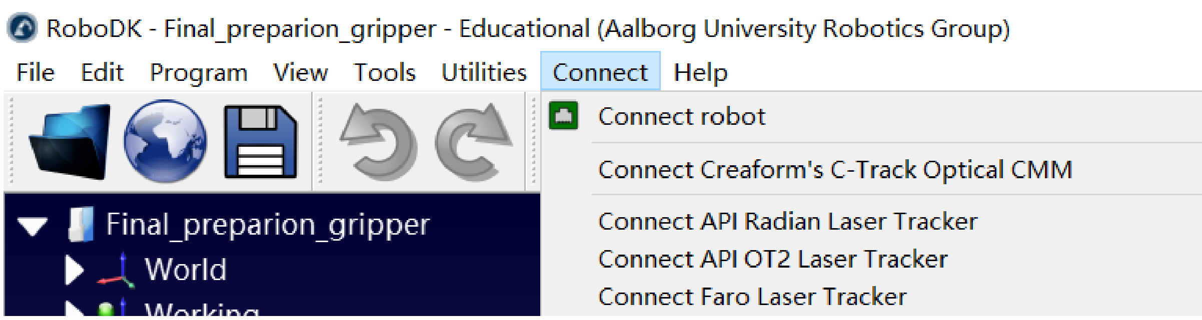


Figure 5.34: How to connect to robot in RoboDK

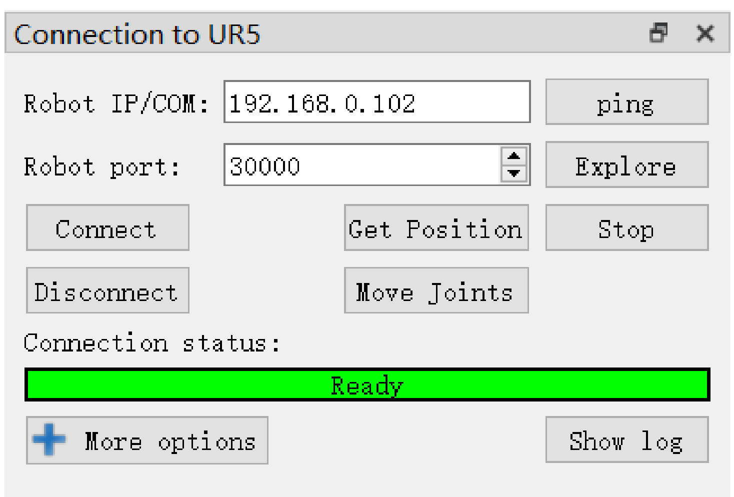


Figure 5.35: The connecting parameters in this project

After connecting the robot, simulate the operation of the software as the last insurance before the actual test. After the check is correct, select the program to be executed by the robot, and right-click '**Send Program to Robot**' to carry out the physical trial run. According to the test results, many debugging will be made, and the final effect can refer to the video demo in GitHub:Automatic-synthesis-of-Molecules-by-UR5. This item has been verified to perform the "operation of transferring test tubes" without errors. In addition, because the posture and size of the syringe are pretty different from the gripper, another path planning is also done in this project.

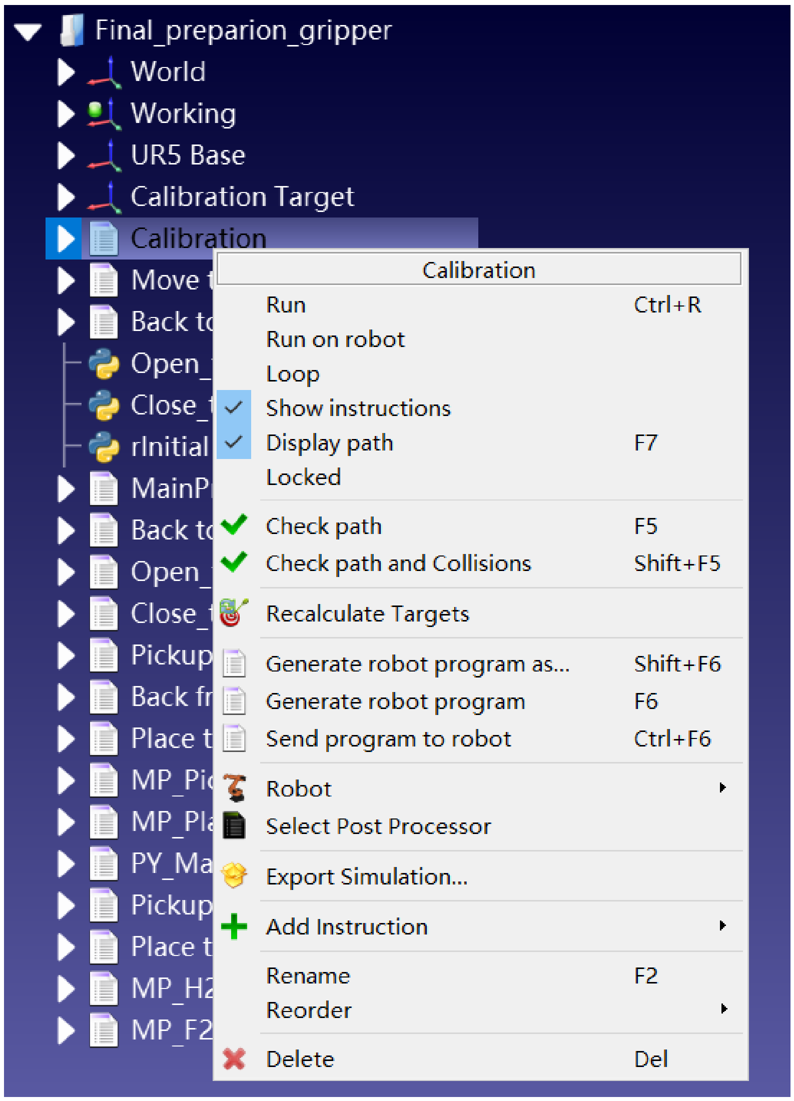


Figure 5.36: How to send the program to the robot

Part V

Conclusion

Chapter 6

Conclusion

The project is expected to replace the traditional human chemical operation with high repetition through the automatic action of one robot to improve experimental efficiency and human safety. This chapter summarises and evaluates the completed parts of the project.

Conclusion

In the task analysis, the ultimate task goal is defined, that is, to try to use robots to perform routine chemical experiments. Based on the fact that the project is still in its early stage, the objectives of this project are set as follows: 1. Use the robot to perform some actions with high repetition in chemical experiments and record them as programs; 2. Integrate these actions into a simulation software in the form of a modular subprogram together with the control program of the tool; 3. The operator can customise the execution sequence of subprograms according to the experimental needs and finally summarise them into the main program. In the analysis of the chemical part, the layout requirements of the robot workbench are sorted out based on visiting the chemical laboratory and recording the shape of common instruments and the size of the workbench. Secondly, decouple the watched complete dilution experiment of quinine solution, and finally, make a one-to-one correspondence between the decoupled essential operation and the robot action. The existing robots and tools in the AAU laboratory are introduced in detail in the technical analysis.

In the simulation stage, this project gives two different concepts. The difference is the number of robots used. Because the UR5 robot is used in this experiment, and its actual workspace is much smaller than the standard workbench in the chemical laboratory, the idea of using two robots to place them relatively came into being. In this chapter, the workbench layout is designed in the simulation software for the two concepts, and the two layout concepts fully meet the requirements put forward by the chemical laboratory. In summary, the two concepts are compared from the aspects of cost performance and operation difficulty. Although both have their advantages and disadvantages, the idea of one robot with a lower cost is retained.

In the practice stage, this project has established two sets of workstations in the simulation software, corresponding to two different tools, respectively. The gripper's control is easier to integrate into the software. The common parts of the two workstations include establishing simulation environments, the setting of the multi-level frame, the position parameter measurement of workbench layout, and the size measurement of instruments and tools. Based on this, 2D three views figures and 3D models are drawn. The difference is that the two tools' size leads to path planning incompatibility, so the relevant subprograms of path planning are made according to the basic operations in task analysis.

Moreover, the robot has tested these programs, and the coordinates of the way-points have been verified to be accurate. In addition, the control method of the gripper has also been integrated into RoboDK, which means that the operator can call the control program anywhere in the main program. This process meets the intuitiveness of operation. The staff with zero-robot-operation-foundation can quickly master establishing the main program in a few minutes. However, due to the structure and control program of the syringe, it can not be integrated into software at present. This report puts forward two feasible methods for this situation. Since both methods require a lengthy research time, it is only used as a reference for follow-up research.

According to the simulation and actual test, it can be concluded that the project has completed the following objectives: the selection of concept, the establishment of the simulation, the methodology of subprogram creation and the integration of the gripper control program into the software. The final effect is that the staff with zero-robot-operation-experience can quickly master the method of making the main program in a short time, and the process is intuitive and straightforward. The main programs that can be executed include the path planning of the syringe and the robot controlling the gripper to move the test tube from any position to the destination.

Evaluation

The final idea of the project is to replace the traditional manual operation with robots completely. However, because of the project's progress, only the initial part is completed, focusing on the integration of control procedures into a single software.

Since the chemical experiment referred to in this project is relatively basic and only includes the use of two types of tools, there are few kinds of robot actions involved. It can be divided into robot movement, gripper opening and closing, and syringe control. According to the starting point and end-point, the robot's movement can be further divided into several subprograms. The experiment aims to establish these subprograms in the simulation and combine them in a modular way to generate the main program that can perform specific tasks. The system can automatically generate the main program into code and send it to the robot.

In the concept proposal stage, two concepts are formulated according to the requirements put forward by the chemical laboratory. Both ideas can meet the project requirements, and the feasibility of this process has been verified in the simulation process. Therefore, according to their advantages and disadvantages, this project chose the concept of a single robot. However, this does not mean that the concept of the two robots will no longer be considered. Given that more and more complex tools and operations will be added in the future, the advantage that the two robots can assist each other may be more prominent.

In the practical part, the robot's path planning and gripper control have been integrated into the software and can be called arbitrarily. This part has been verified. The function of being improved is the control of the syringe. Since the syringe is controlled by Arduino and independent of the robot, the software cannot control the syringe in theory. However, there are still two methods to solve this problem. One is to redesign the physical structure of the syringe and bind the input trigger condition of the controller to the output of the robot. At this time, the syringe can be controlled indirectly only by controlling the output on the robot flange with software. Second, the original C + + program should be wholly rewritten into a python program. This is because only the Python program can be compatible with both Arduino and RoboDK. In addition to the above two methods, it does not rule out the existence of other more straightforward solutions, which need to be further studied in future work.

In conclusion, the project has accurately completed most of the initial goals and has been verified to be feasible. Due to time constraints, the remaining part needs to be continuously developed in future work.

Chapter 7

Future Works

This project has completed the two fundamental problems of robot path planning and gripper control and successfully combined them in RoboDK software. The easy-to-use feature enables even non-robot practitioners (Chemical operators) to master how to control the robot quickly. However, the project is still in its early stage, with a lot of work still placed on the to-do list. This chapter will list some future work and future research directions.

Future Work

The future work is mainly to repair the deficiencies in the current project. According to the problems mentioned above, including the inability to control the syringe, the inability to write the syringe program into robodk software, the setting of the laboratory is not enough to restore the chemical laboratory, and the model of the robot is not enough to cover all areas of the workbench, etc. This section will focus on the direction of improvement and possible successful methods.

1. Modify the physical structure of the syringe. The syringe is independent of the robot at this stage, so the software cannot connect and control the syringe. The first solution is to change the input signal of the syringe to the digital output or analogue output of the robot. When the robot receives the program, it will release the current signal, which will become the input trigger signal of the syringe control board. Finally, the software can indirectly control the push and pull of the syringe handle;
2. Rewrite the control procedure of the syringe. RoboDK currently only supports Python programs, while syringes use C + + programs. These two are incompatible. Therefore, try to rewrite the code without changing the physical conditions of the syringe. Through 'firmata' or 'Pyfirmata', Arduino can be compatible with Python;
3. Expand the experimental workbench. The expected size of the chemical laboratory is 2m × 1.2m. The current project cannot meet the demand, so it is suggested to expand the experimental workbench;
4. Add experimental instruments. There are too few experimental instruments in the current project, which makes it impossible to restore a natural chemical experiment accurately;
5. Change the robot. The current working range of UR5 is far less than the expected size, so it can be considered to replace a larger model of a robot;

6. Establish a more prosperous simulation environment, and the method is almost the same as that of this project;
7. Redesign the placement and replacement of tools

Future Research

As the project is still in its early version, there are still a lot of research directions for follow-up development. Only some suggestions are listed here for reference:

1. Increase the types of chemical experimental instruments to meet the requirements of more complex chemical experiments;
2. Standardize the trigger conditions of the end tools of the robot and generalise the control program of the tools. For example, all tools can be controlled by controlling the output of the robot;
3. Optimize the controller of the end tool of the robot, preferably using a unified program language, such as Python;
4. Simplify the working interface of the software deeply. In the current RoboDK workstation, the main program and subroutine are not classified in the structure tree, resulting in that the operator needs to select the required subroutine from a large number of parallel programs every time to form a new main program;
5. Try to bring the preparation stage of the chemical experiment into the operating range of the robot;
6. Add a 3D camera to obtain the position of the object through the machine vision algorithm instead of the fixed position;
7. The subroutine can be further refined and decoupled into the form of **Tool + Action + Object + Place adverbial**. They are combined in the form of **puzzle blocks** to achieve a broader range of action coverage.

Bibliography

[1] Katharine Sanderson. "Automation: Chemistry shoots for the Moon". In: *Nature* 568(7753).577-579 (2019). DOI: 10.1038/d41586-019-01246-y..

[2] Csaba Urbaniczky Christian Bernlind. "An Efficient Laboratory Automation Concept for Process Chemistry". In: *Organic Process Research Development* 13.1059–1067 ().

[3] Luc Van Hijfte Gilbert Marciniak Nicolas Froloff. "Combinatorial chemistry, automation and molecular diversity: new trends in the pharmaceutical industry". In: *Journal of Chromatography B: Biomedical Sciences and Applications* 725.3-15 ().

[4] Elizabeth Farrant. "Automation of Synthesis in Medicinal Chemistry: Progress and Challenges". In: *ACS Medicinal Chemistry Letters* 2020, 11, 8.1506-1513 (). DOI: 10.1021/acsmedchemlett.0c00292.

[5] Stockio.com. *Free icon "Chemistry icon"*. https://www.stockio.com/free-icon/beatriz-icons-chemistry. (Accessed on 28/05/2022). 2022.

[6] Freepik. *Robotic arm - Free industry icons*. https://www.flaticon.com/free-icon/robotic-arm_1839269. (Accessed on 28/05/2022). 2022.

[7] Freepik. *Syringe - Free medical icons*. https://www.flaticon.com/free-icon/syringe_2770407. (Accessed on 28/05/2022). 2022.

[8] wiredworkers. *Meet Universal Robots UR5*. https://wiredworkers.io/universal-robots-ur5/. (Accessed on 20/05/2022). 2022.

[9] Universal Robots A/S. *UNIVERSAL ROBOT UR5e*. https://www.universal-robots.com/products/ur5-robot/. (Accessed on 20/05/2022). 2022.

[10] Universial Robot. *RG2 – FLEXIBLE 2 FINGER ROBOT GRIPPER WITH WIDE STROKE*. https://www.universal-robots.com/plus/products/onrobot/rg2-gripper/. (Accessed on 20/05/2022). 2022.

[11] OnRobot. *RG2 Gripper - Flexible 2 Finger Robot Gripper | OnRobot*. https://onrobot.com/en/products/rg2-gripper. (Accessed on 20/05/2022). 2022.

[12] Association for Advancing Automation. *RoboDK: A simulator for industrial robots robot programming | Member of A3*. https://www.automate.org/companies/robodk. (Accessed on 21/05/2022). 2022.

[13] RoboDK. *Simulator for industrial robots and offline programming - RoboDK*. https://robodk.com/. (Accessed on 21/05/2022). 2022.

[14] Benjamin C. Kuo. *Automatic Control Systems (6th ed.)* New Jersey: Prentice Hall., 1991. ISBN: 0-13-051046-7.

[15] Mofeywalker. *Support for OnRobot RG2 Gripper by Mofeywalker · Pull Request 45 · SintefManufacturing/python-urx · GitHub*. https://github.com/SintefManufacturing/python-urx/pull/45/files. (Accessed on 24/05/2022). 2022.

[16] MartinJensen37. *Automated$_s$ynthesis*. https://github.com/MartinJensen37/Automated_synthesis/blob/main/ros_pkgs/ur5_interface/scripts/service_controller.py. (Accessed on 28/05/2022). 2022.